REMARKS

Applicants submit the following comments as a summary of the personal interview on October 3, 2001, and as a follow up to the Amendment filed November 5, 2001, in response to the outstanding matters. Additionally, the title of the invention has been amended.

Extension of Time

A Petition has been filed under the provisions of 37 CFR §1.136 for an extension of time to respond to the Examiner's Action of February 11, 2002. The appropriate fee set forth in 37 CFR § 1.17 is filed herewith.

Discussion of Space Occupied Rate and other features

As requested by Examiner Sheinberg, the following comments are directed to a working example to explain the "spaced occupied rate" and other questions discussed with the Examiner. The working example is in line with claim 1.

For the sake of convenience the sample molecule is a three atom model and is composed of three atoms 2a(C1), 2b(C), 2c(C) as C1-C=C. For the sake of convenience, the sample molecule is reduced into the two dimensional plane.

(A) In items (1), (2), (3) and (4) of the attached sketches, the setting of the molecule surrounding surface, the

dividing of the molecule surrounding space into a plurality of component spaces, assuming of the frontier surrounding surface, and providing of the probe points on the frontier surrounding surface are shown.

(B) Deriving of the space occupied rate (FFfield) is shown in the item (5) of the attached papers.

The space occupied rate (FFfield) is given as the number of the probe points existing on the frontier spherical surface of each of the atoms 2a, 2b, 2c.

Though both of the atoms 2b and 2c are in fact carbon atoms, it is assumed for the sake of convenience that all are probe atoms. Additionally, for the sake of convenience, all of the probe points are shown in the figure of item (5). FFfield is for the atom 2a = 7, FFfield is for the atom 2b = 3 and FFfield is for the atom 2c = 8 are obtained.

The fact that FFfield of the atom 2a or the atom 2c is much larger than FFfield of the atom 2b shows that the proportion occupied by the atom 2a or the atom 2c in the whole reaction characteristic of the molecule is much larger than that of the proportion occupied by the atom 2b.

The fact that FFfield of the atom 2a and FFfield of the atom 2c are not so different shows that the proportions occupied by the atom 2a and the atom 2c in the whole reaction characteristic of the molecule are almost the same.

(C) Deriving of the electrostatic factor (FFelectro) is shown in the item (6) of the attached sketches.

In order to derive the electrostatic factor (FFelectro), the electrostatic interaction energy between a unit charge set at the probe point(i) and the charges (charge(2a), charge(2b) and charge(2c)) of the atoms 2a, 2b, 2c is obtained, and the electrostatic energy thus obtained is summed up all over the probe points on the frontier surface of the atom 2a, and FFelectro of the atom 2a is obtained.

FFelectro of the atoms 2b, 2c is also obtained in the same way.

In fact, the electrostatic factor (FF electro) of the one atom, for example of the atom 2a, is large which shows that a molecule, which is to be reacted with the present molecule, is difficult to approach to the atom 2a of the present molecule if the molecule has a positive charge and the molecule is easy to approach to the atom 2a, if the molecule has a negative charge.

(D) Deriving of the steric factor (FFsteric) in the item (7) of the attached papers.

In order to derive the steric factor (FFsteric), the probe atom sp′ carbon is set at the probe point (i). The probe atom sp′ carbon is set on the probe point so as that the van der Waals radius of the probe atom sp′ carbon is in contact with the probe point (i). van der Waals energies (=steric interaction energies)

between the probe atom sp' carbon set at the probe point (i) and the atoms 2a, 2b, 2c are obtained by using a mathematical technique for calculating a van der Waals energy in a (well known) molecular force field MM3. The van der Waals energy thus obtained is summed up all over the probe points on the frontier surface of the atom 2a, and FFsteric of the atom 2a is obtained.

FFsteric of the atom 2b, 2c is also obtained in the same way.

The fact that the steric feature (FF steric) of the one atom, for example, of the atom 2a, is large shows that a molecule, which is to be reacted with the present molecule, is difficult to approach to (or 'attack to') the atom 2a of the present molecule since the steric hinderance due to the atom 2a is large.

(E) In order to numerically characterize the electronic and steric factors of the molecule, three kinds of features (space occupied rate (FF field), electrostatic feature (FF electro) and steric feature (FF steric)) are obtained for the atom 2a. The three kinds of features are also obtained for the atoms 2b, 2c.

Attached hereto as exhibits are the requested paper of a Voronoi division. It is reiterated that the Voronoi division is not essential for the present invention, because the Voronoi division is only one embodiment of a procedure for spatially dividing the molecule surrounding space 21 into the plurality of

component spaces. For the Examiner's reference, Fig.31 on page 380 of the document shows similar procedure for dividing a space.

It is respectfully submitted that the comments set forth above and those comments in the Amendment filed November 5, 2001 fully respond to the questions raised during the interview on October 3, 2001.

Conclusion

In view of the foregoing, it is believed all the issues raised by the Examiner have been considered and appropriately addressed. It is believed this application is now in condition for allowance and action to that end is respectfully solicited.

If the Examiner believes that a further conference would be of value in expediting the prosecution of this application, the Examiner is invited to telephone the undersigned to arrange for such a conference.

Attached hereto is a marked-up version of the changes made to the title by the current amendment. The attached page is captioned "**Version with markings to show changes made.**"

Respectfully submitted,
JACOBSON HOLMAN PLLC

By _____
John C. Holman
Reg. No. 22,769

400 Seventh Street, N.W.
Washington, D.C. 20004-2201
(202) 638-6666
Date: April 15, 2002
Atty. Docket: 5970/P63431US0
JCH:DKD:VJB

Version with markings to show changes made.

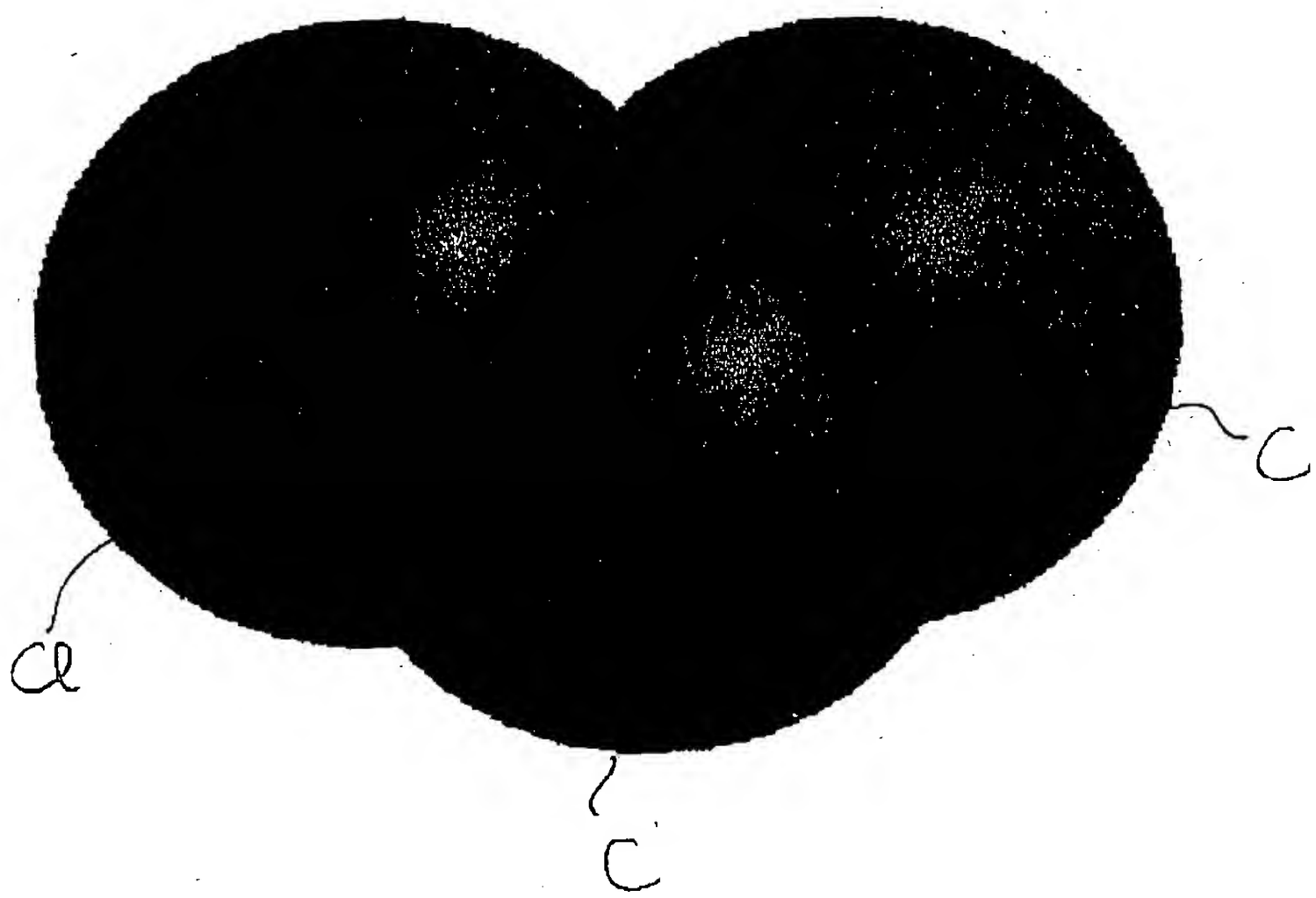
In the Title:

Please amend the title as follows:

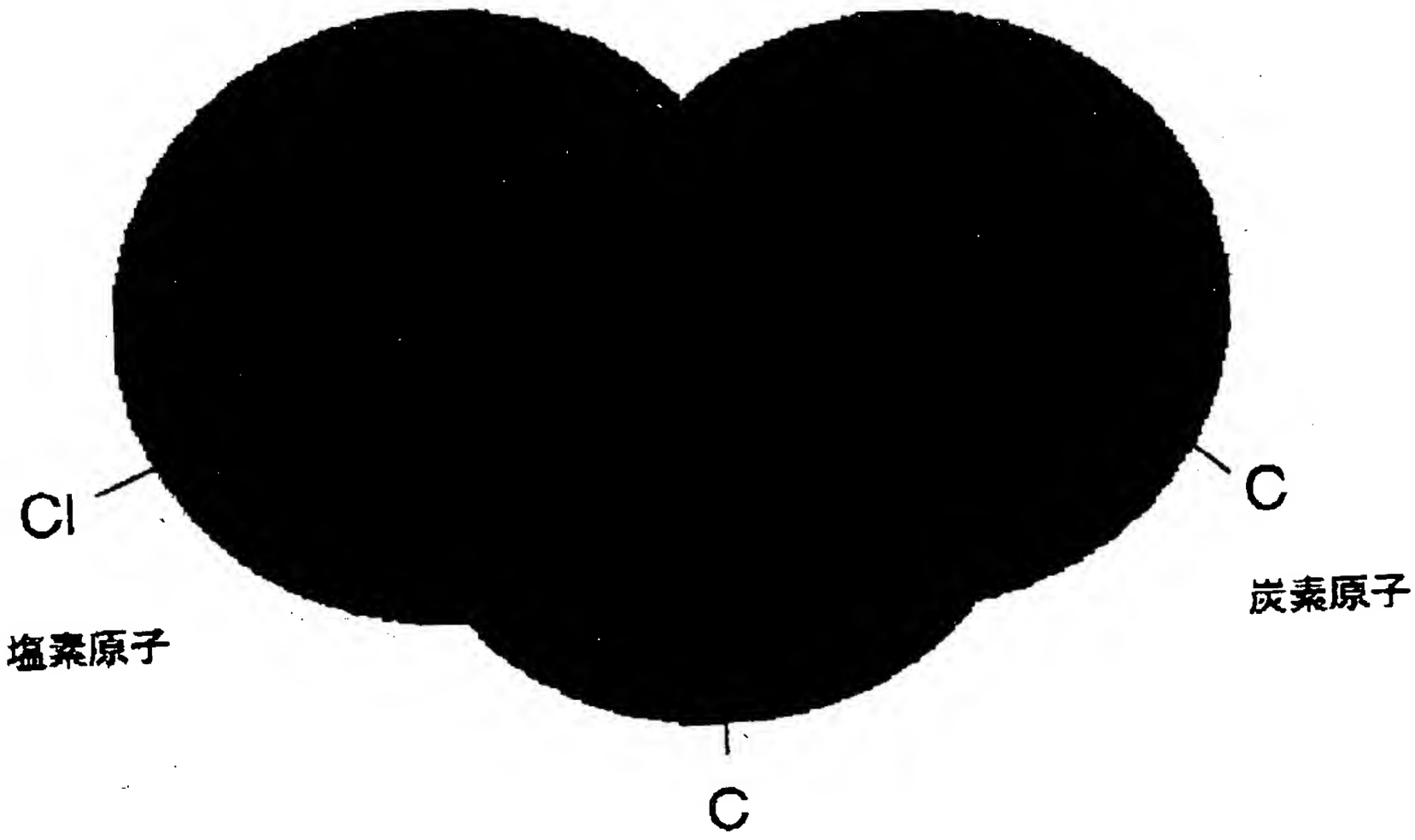
--METHOD FOR PREDICTING [FEATURES OF CHEMICAL] REACTION CHARACTERISTICS OF MOLECULES--

09/266,813
#15

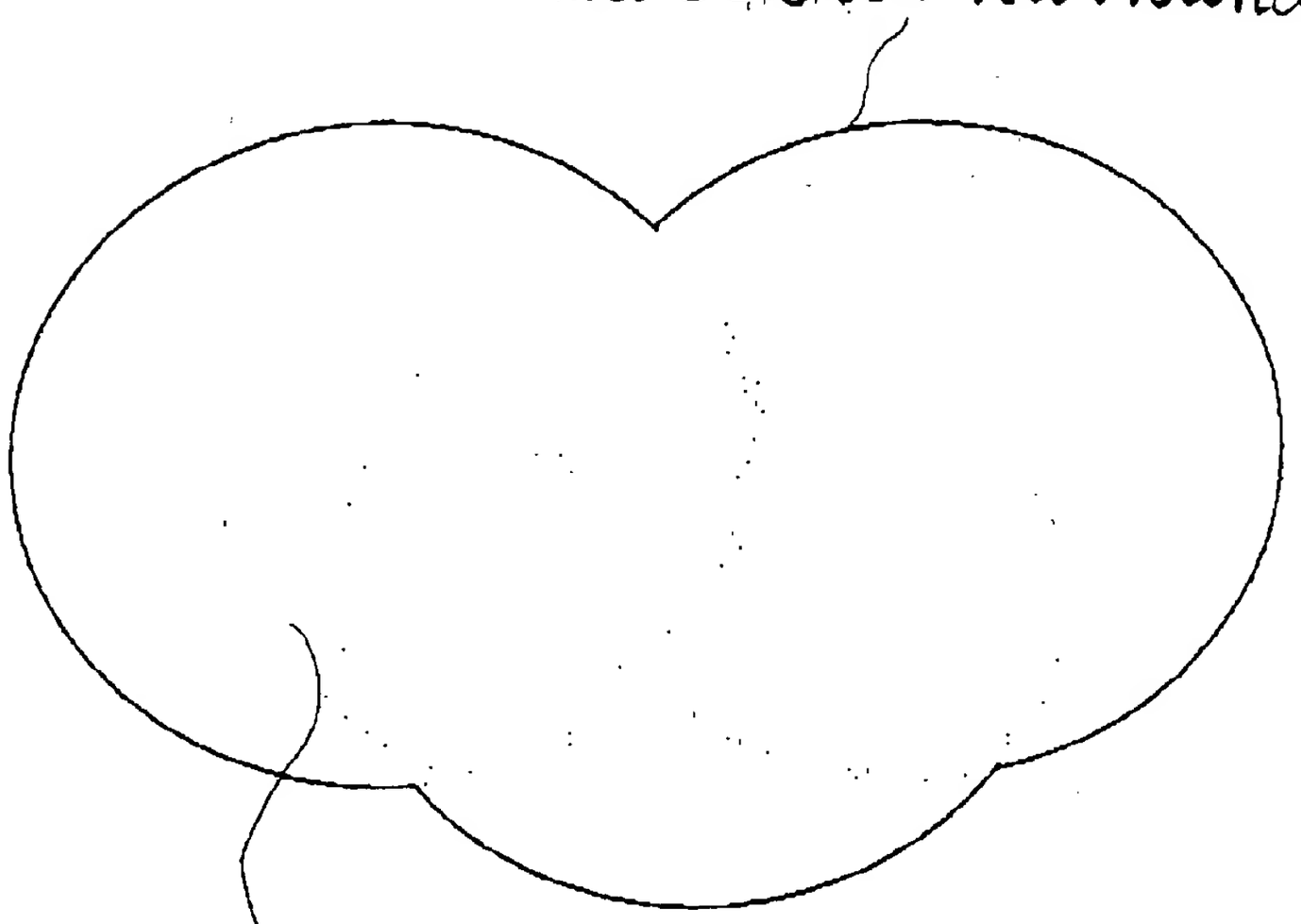
分子包囲面を描いた図



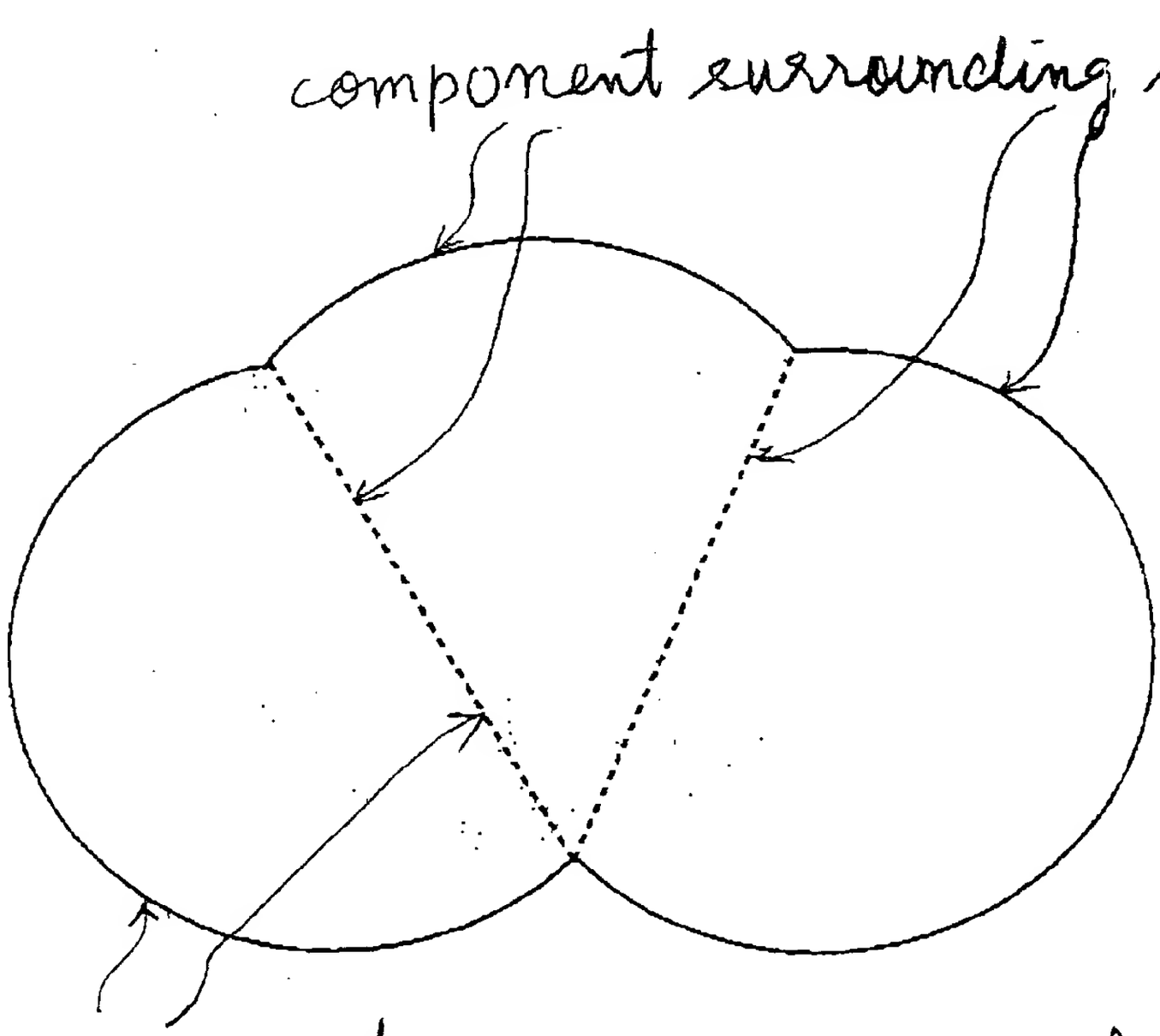
モデルを構成する原子種



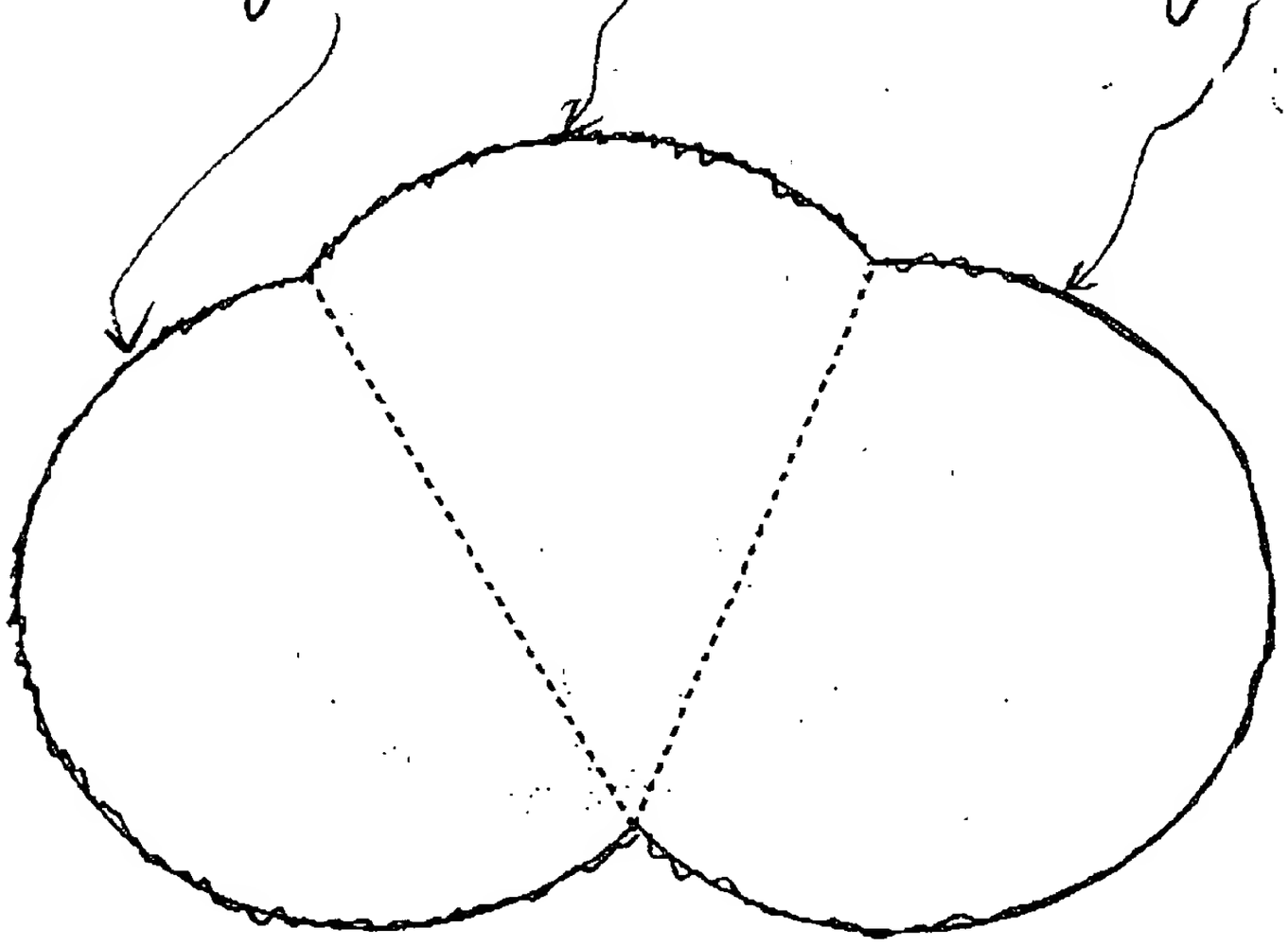
(1) setting of molecular surrounding surface.



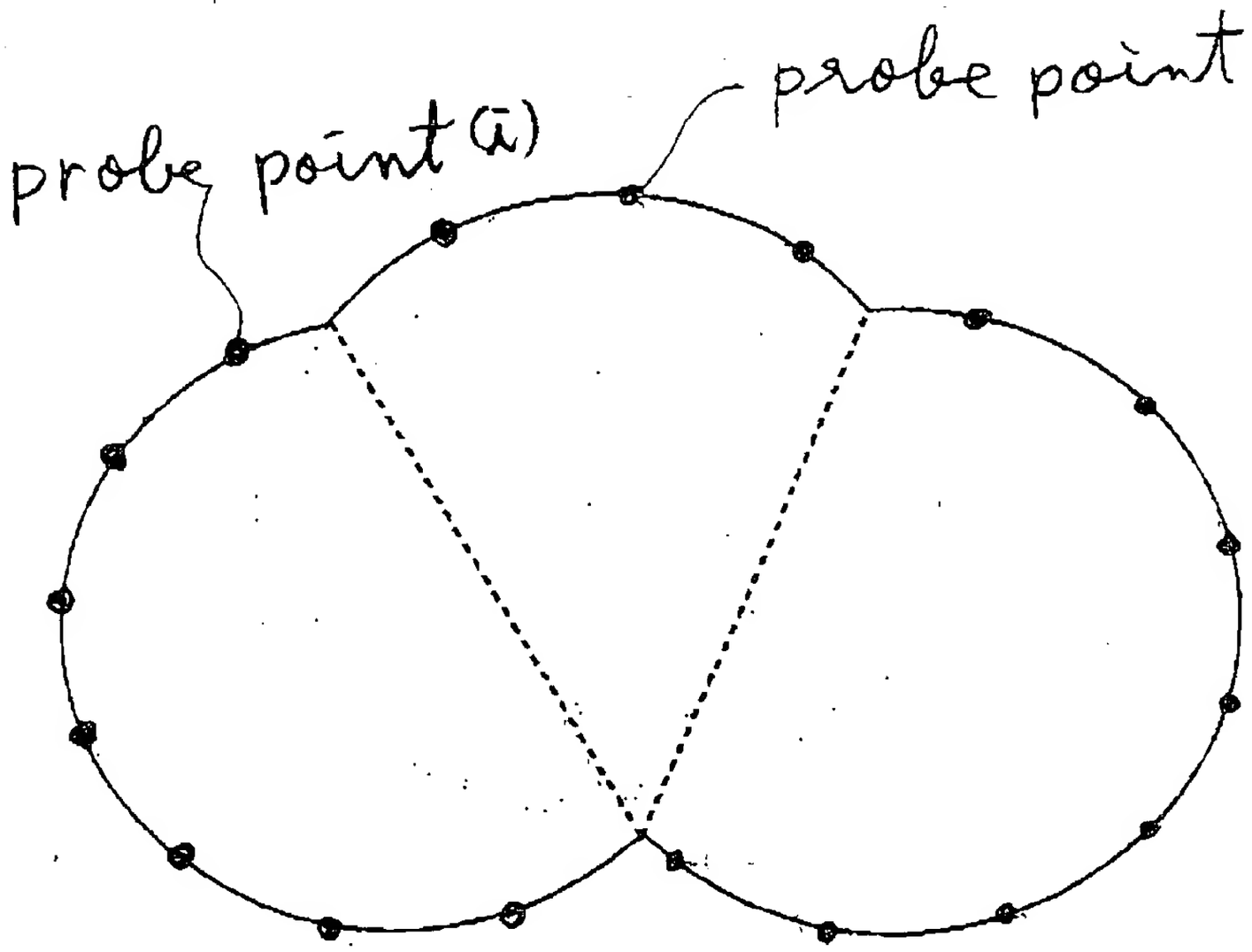
(2) dividing of molecular surrounding space



(3) assuming of frontier surrounding surface



(4) providing of probe points

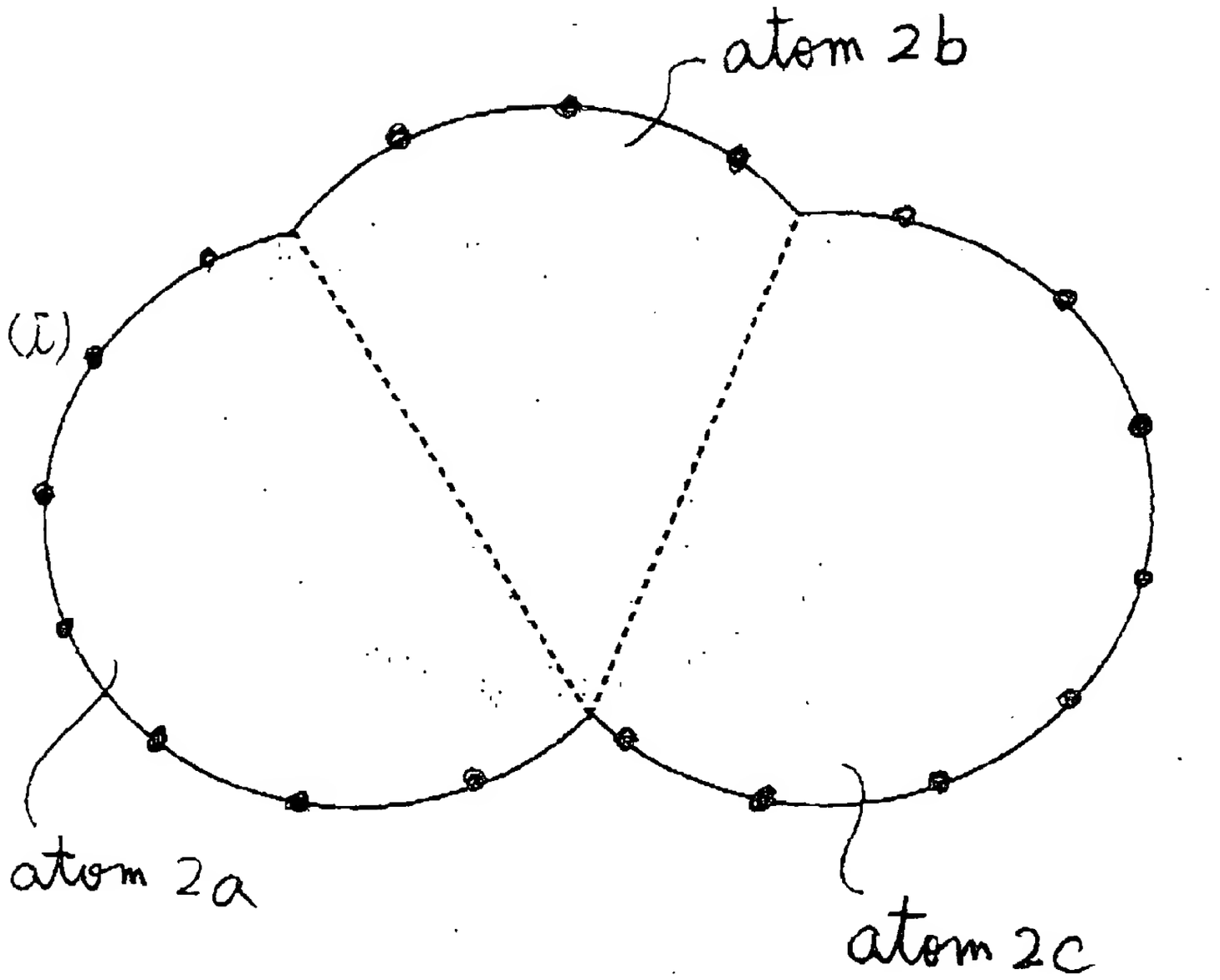


(b) deriving of space occupied rare (FF field)

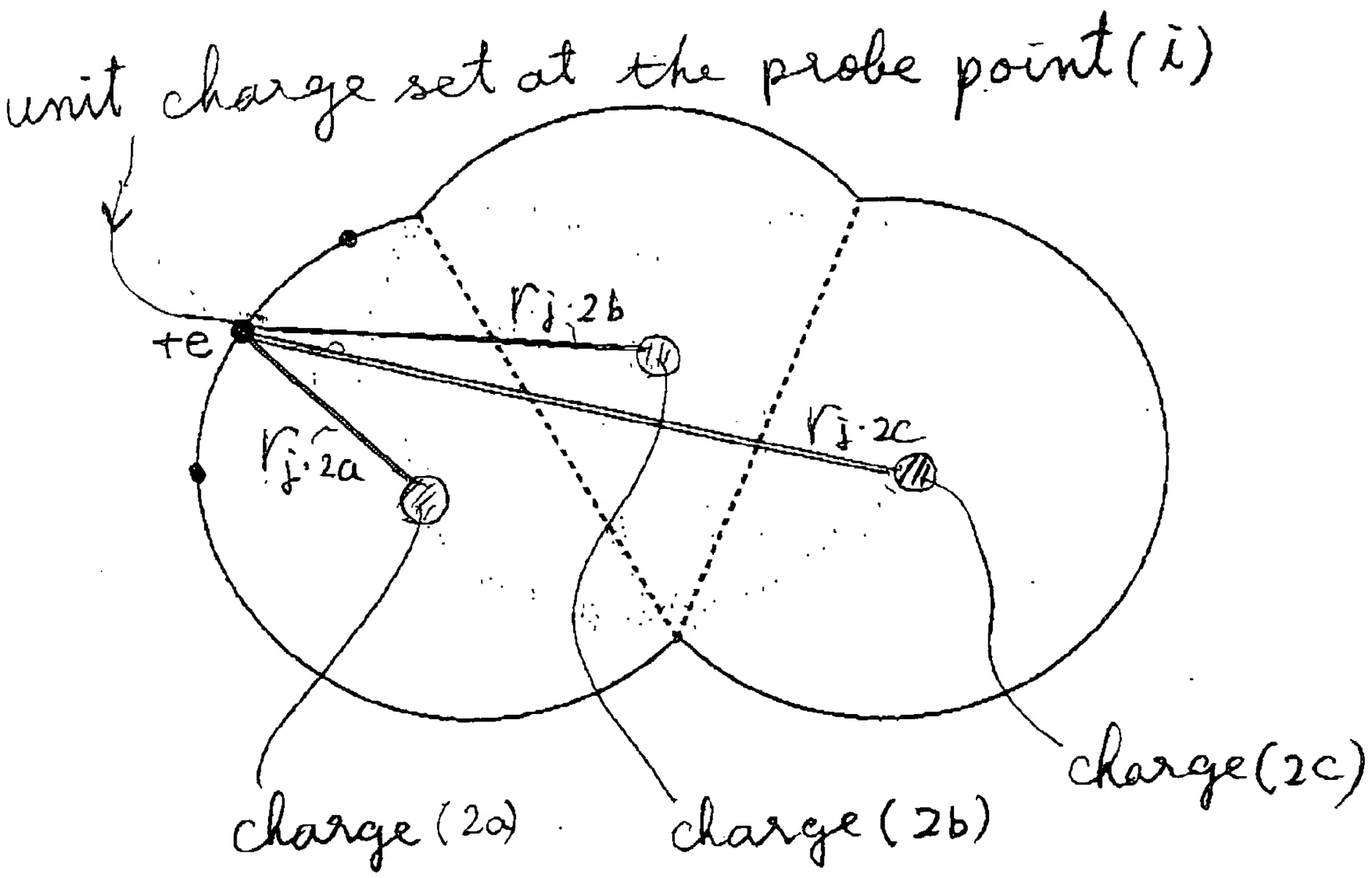
FF field for atom 2a = 7

FF field for atom 2b = 3

FF field for atom 2c = 8

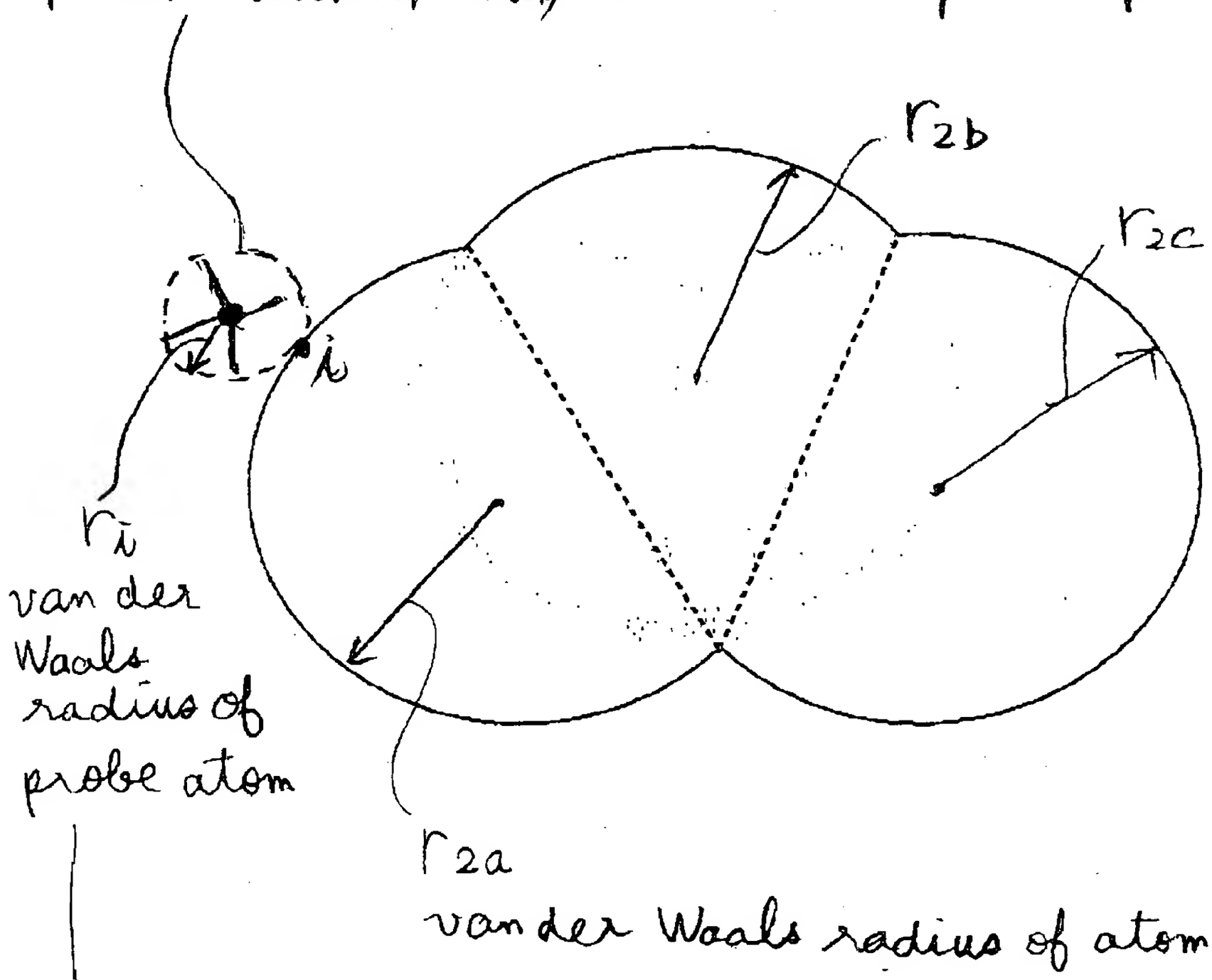


(6) deriving of electrostatic factor
(FFelectro)



(7) deriving of steric factor (FF: steric)

probe atom (sp^3 carbon) set at the probe point (i)



EXHIBIT

#1

Voronoi Diagrams — A Survey of a Fundamental Geometric Data Structure

FRANZ AURENHAMMER

Institute für Informationsverarbeitung Technische Universität Graz, Schießstattgasse 4a, Austria

This paper presents a survey of the Voronoi diagram, one of the most fundamental data structures in computational geometry. It demonstrates the importance and usefulness of the Voronoi diagram in a wide variety of fields inside and outside computer science and surveys the history of its development. The paper puts particular emphasis on the unified exposition of its mathematical and algorithmic properties. Finally, the paper provides the first comprehensive bibliography on Voronoi diagrams and related structures.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*geometrical problems and computations*; G.2.1 [**Discrete Mathematics**]: Combinatorics—*combinatorial algorithms*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*geometric algorithms, languages, and systems*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Cell complex, clustering, combinatorial complexity, convex hull, crystal structure, divide-and-conquer, geometric data structure, growth model, higher dimensional embedding, hyperplane arrangement, k-set, motion planning, neighbor searching, object modeling, plane-sweep, proximity, randomized insertion, spanning tree, triangulation

INTRODUCTION

Computational geometry is concerned with the design and analysis of algorithms for geometrical problems. In addition, other more practically oriented, areas of computer science—such as computer graphics, computer-aided design, robotics, pattern recognition, and operations research—give rise to problems that inherently are geometrical. This is one reason computational geometry has attracted enormous research interest in the past decade and is a well-established area today. (For standard sources, we refer to the survey article by Lee and Preparata [1984] and to the textbooks by Preparata and Shamos [1985] and Edelsbrunner [1987b].)

Readers familiar with the literature of computational geometry will have noticed, especially in the last few years, an increasing interest in a geometrical construct called the *Voronoi diagram*. This trend can also be observed in combinatorial geometry and in a considerable number of articles in natural science journals that address the Voronoi diagram under different names specific to the respective area. Given some number of points in the plane, their Voronoi diagram divides the plane according to the *nearest-neighbor*

CONTENTS

rule: Each point is associated with the region of the plane closest to it; (Figure 1).

Why do Voronoi diagrams receive so much attention? What is special about this easily defined and visualized construct? It seems three main reasons are responsible. First, Voronoi diagrams arise in nature in various situations. Indeed, several natural processes can be used to define particular classes of Voronoi diagrams. Human intuition is often guided by visual perception. If one sees an underlying structure, the whole situation may be understood at a higher level. Second, Voronoi diagrams have interesting and surprising mathematical properties; for instance, they are related to many well-known geometrical structures. This has led several authors to believe that the Voronoi diagram is one of the most fundamental constructs defined by a discrete set of points. Finally, Voronoi diagrams have proved to be a powerful tool in solving seemingly unrelated computational problems and therefore have increasingly attracted the attention of computer scientists in the last few years. Efficient and reasonably simple techniques have been developed for the computer construction and representation of Voronoi diagrams.

The intention of this survey is threefold: First, motivated by the fact that Voronoi diagrams have been (re)invented and studied fairly independently in the applied natural sciences, in mathematics, and in computer science, it presents sketches of their historical development in these three areas. Second, it surveys the literature on Voronoi diagrams and related structures, with particular emphasis on the unified exposition of their mathematical and computational properties and their applications in computer science. Finally, it provides the first comprehensive bibliography on Voronoi diagrams.

Basic Properties of the Voronoi Diagram

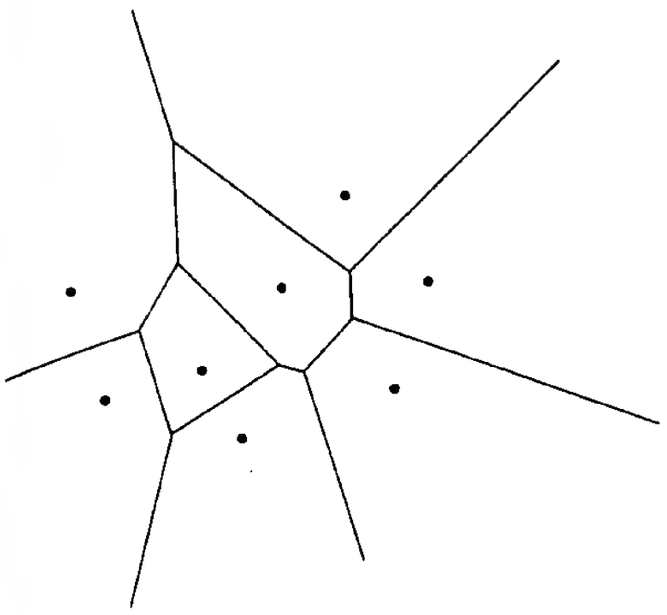
We begin with a description of elementary, though important, properties of the Voronoi diagram that will suggest some feeling for this structure. We also introduce notation used throughout this paper. See also Preparata and Shamos [1985] or Edelsbrunner [1987] for sources on this material.

We first give a usual generic definition of the Voronoi diagram. Let S denote a set of n points (called *sites*) in the plane. For two distinct sites $p, q \in S$, the *dominance* of p over q is defined as the subset of the plane being at least as close to p as to q. Formally,

$$\mathrm{dom}(p, q) = \{ x \in R^2 \mid \delta(x, p) \le \delta(x, q) \},$$

for δ denoting the euclidean distance function. Clearly, $\mathrm{dom}(p, q)$ is a closed half plane bounded by the perpendicular bisector of p and q. This bisector separates all points of the plane closer to p from those closer to q and will be termed the *separator* of p and q. The *region* of a site $p \in S$ is the portion of the plane lying in all of the dominances of p over the remaining sites in S. Formally

$$\mathrm{reg}(p) = \bigcap_{q \in S - \{p\}} \mathrm{dom}(p, q).$$



1. Voronoi diagram for eight sites in the

ce the regions are coming from in-
:ting $n-1$ half planes, they are
x polygons. Thus the boundary of a
ı consists of at most $n-1$ *edges*
mal open straight-line segments)
ertices (their endpoints). Each point
ı edge is equidistant from exactly
ites, and each vertex is equidistant
at least three. As a consequence,
:gions are edge to edge and vertex
rtex, that is to say, they form a
ɔnal partition of the plane. This
ion is called the *Voronoi diagram*,
of the finite point-set S (Figure 1).
e that a region, say reg(p), cannot
ıpty since it contains all points of
ane at least as close to p as to any
sites in S. In particular, $p \in \text{reg}(p)$.
ows that $V(S)$ contains exactly n
s. Some of them are necessarily
nded. They are defined by sites ly-
the boundary of the convex hull of
ıuse just for those sites there exist
arbitrarily far away but still clos-
No vertices occur if and only if all
n S lie on a single straight line.
legenerate configurations also im-
: existence of regions with only one
ınded) edge. Otherwise, three or
:dges meet at a common vertex. It

ıvex hull of S is the smallest convex poly-
contains S.

should be observed that each vertex is the center of a circle that passes through at least three sites but encloses no site.

Although n sites give rise to $\binom{n}{2} = O(n^2)$ separators, only linearly many separators contribute an edge to $V(S)$. This can be seen by viewing a Voronoi diagram as a planar graph with n regions and minimum vertex degree 3. Each of the e edges has two vertices, and each of the v vertices belongs to at least three edges. Hence, $2e \geq 3v$. Euler's relation $n + v - e \geq 2$ now implies $e \leq 3n - 6$ and $v \leq 2n - 4$. Thus, for example, the average number of edges of a region does not achieve six; there are less than $3n$ edges, and each of them belongs to exactly two of the n regions.

The linear behavior of the size of the Voronoi diagram in the plane means that, roughly speaking, this structure is not much more complex than the underlying configuration of sites. This is one of the main reasons for the frequent use of Voronoi diagrams. A second reason is that $V(S)$ comprises the entire proximity information about S in an explicit and computationally useful manner. For example, its applicability to the important post-office problem (see below) is based on the trivial observation that a point x falls into the region of a site p if and only if p is closest to x among all sites in S. Moreover, if site p is closest to site q, then reg(p) and reg(q) share a common edge. This particularly implies that the closest pair of sites in S gives rise to some edge of $V(S)$.

Applications in Computer Science

To substantiate the usefulness of the Voronoi diagram in computer science, we briefly describe four situations where this structure is used. The practicality and diversity of these applications will impart the appeal of Voronoi diagrams.

Associative File Searching

Consider some file of n two-attribute records referring, for example, to latitude and longitude of a city or to age and

income of a person. Suppose we are given an additional target record R and we quickly want to retrieve that record of the file that matches R the best. If retrieval occurs frequently on the same file, a supporting data structure is called for. This *associative file searching* problem was first posed by Knuth [1973] as a two-dimensional generalization of the usual (one-attribute) file-searching problem. It possibly is best known in its geometric version under the name post-office problem: Given a set S of n sites in the plane (post offices), report a site closest to a given query point q (the location of a person). Note that there exists a trivial $O(n)$-time solution by computing all n distances. There are various algorithms in computational geometry that need the post office problem as a subroutine [Preparata and Shamos 1985].

Shamos [1975a] first observed the relevance of Voronoi diagrams to this problem. A site p is closest to q if and only if q falls into the region of p. In a preprocessing step the Voronoi diagram of S is computed. To report a site closest to q, it now suffices to determine the region that contains q. For this so-called point-location problem, efficient solutions have been developed by Kirkpatrick [1983], Edahiro et al. [1984], and Edelsbrunner et al. [1986]. In particular, point location in a Voronoi diagram with n regions is supported in $O(\log n)$ time and $O(n)$ storage overhead. This shows that the post-office problem can be solved by means of Voronoi diagrams in logarithmic query time and without increasing the order of space, which is well known to be optimal already for usual file searching.

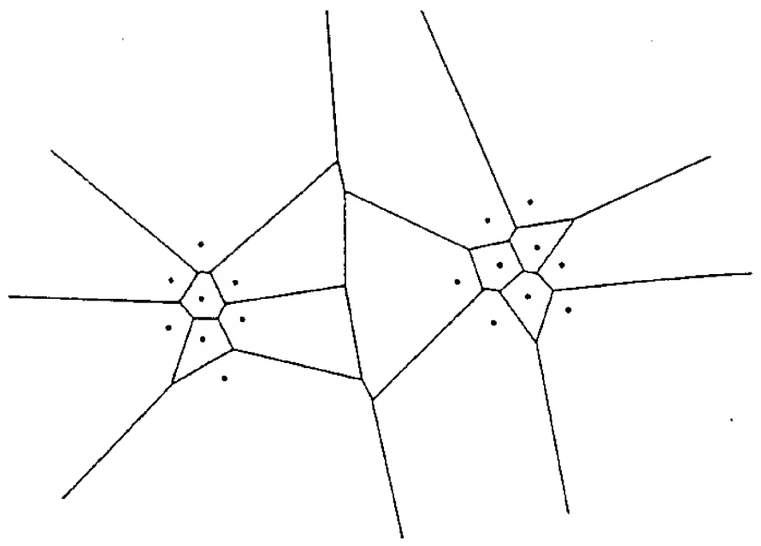


Figure 2. Voronoi diagram for two dense clusters.

Cluster Analysis

The problem of automatically clustering data arises frequently [Hartigan 1975]. Finding clusters means determining a partition of the given set of data into subsets whose in-class members are similar and whose cross-class members are dissimilar according to a predefined similarity measure. In the case of two-attribute data, similarity is reflected by the proximity of sites in the plane. Proximity, in turn, is revealed by properties of the Voronoi diagram for these sites. For instance, dense subsets of sites give rise to Voronoi regions of small area; regions of sites in a homogeneous cluster will have similar geometric shape; for clusters having orientation-sensitive density, the shapes of the regions will exhibit a corresponding direction sensitivity. Figure 2 gives an example. Ahuja [1982] showed how to use these properties for clustering and matching sites.

Voronoi diagrams support various clustering techniques used in practice. What is required at any stage of the clustering process is often little more than the retrieval of the nearest-neighbor sites of specified sites. They can be reported easily by examining the edges of the regions of the specified sites. This applies to several hierarchical methods [Murtagh 1983], partitional methods [Asano et al. 1988], and methods involving cluster selection [Aggarwal et al. 1989].

Scheduling Record Accesses

Consider a mass storage system represented by a two-dimensional array of grid points, each capable of storing one record. The time for the read/write head to move from point (x, y) to the point (u, v) is proportional to $|x - u| + |y - v|$, the distance between these points measured

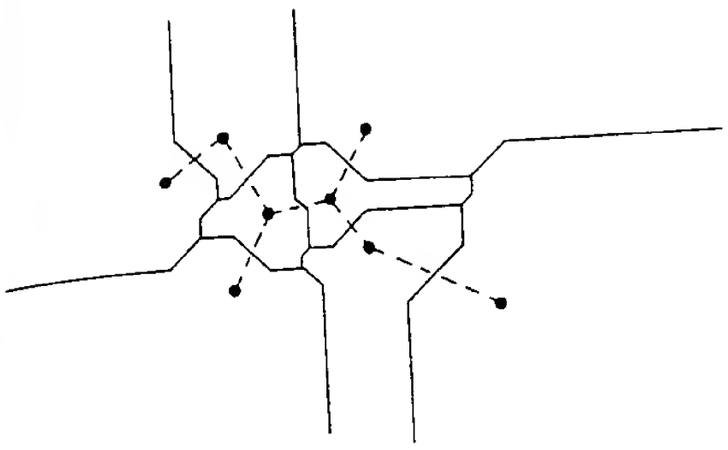


Figure 3. Voronoi diagram and minimum spanning trees in the L_1-metric.

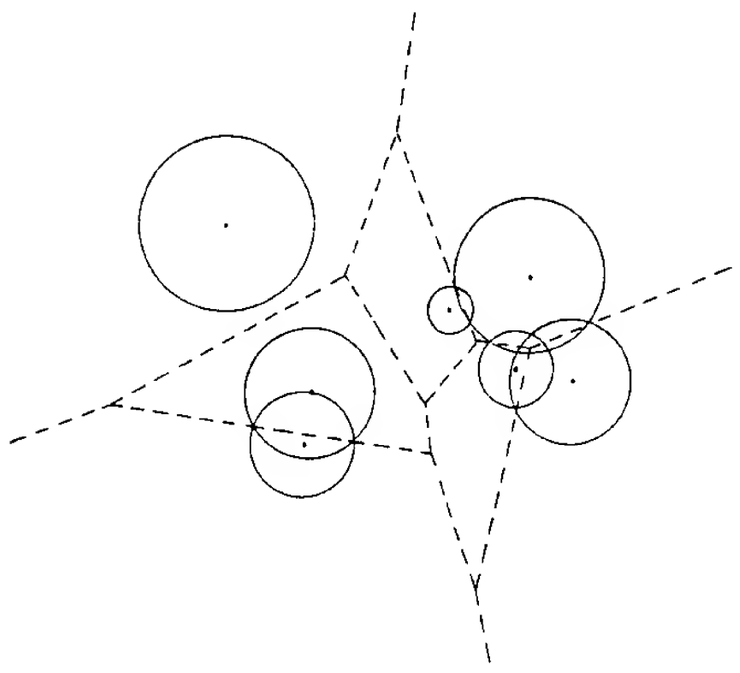


Figure 4. Power diagram for seven circles.

in the L_1-metric (i.e., Manhattan distance). One problem that arises in accessing batched requests is the following: How is the head movement to be scheduled in order to retrieve the requested records in minimum time (distance)?

Finding an exact solution was shown to be NP-complete by Lee and Wong [1980]. A satisfactory approximate solution can be obtained quickly by means of Voronoi diagrams. In an initial step, the Voronoi diagram induced under the L_1-metric is computed in $O(n \log n)$ time by taking the n requested grid points as sites. Now the minimum-L_1-length tree spanned by the sites can be constructed quickly: An edge of the tree can only connect sites whose regions are neighbored in the Voronoi diagram [Hwang 1979; Lee and Wong 1980]. Figure 3 illustrates this diagram (solid) and the corresponding spanning tree (dashed). To obtain a head movement whose length is within a factor of 2 of the optimum, each tree edge is traversed twice to access all the sites.

Collision Detection

An important topic in controlling the motion of robot systems is that of collision detection. For a robot moving in an obstacle environment, one needs to determine collisions between moving robot subparts and stationary obstacles or between two separately moving subparts of the robot. In particular, *proximity detection* is important in order to be able to stop the system before a collision will have occurred.

The robot system and the environment of obstacles are usually modeled by polygonal objects. For the sake of proximity detection, collision-critical points on the boundary of these objects may be circumscribed by circles whose radii correspond to the tolerance threshold of the system. This reduces the problem to detecting the intersection of circles. Colors may be assigned to the circles in order to distinguish between "harmless" intersections (among circles stemming from the same moving robot part or the same obstacle) and others. Sharir [1985] pointed out that there will be no intersection between circles of different colors if all the connected components formed by the circles are unicolored. Finding connected components and checking colors seem to involve an inspection of each pair of circles. The problem becomes easy, however, if the power diagram (the Voronoi diagram where the power lines of the circles are taken as separators) is available; see Aurenhammer [1988a] where this problem is solved in $O(n \log n)$ time for n colored circles. The crucial property that can be exploited is that the points of intersection of two circles lie on their common power line (Figure 4).

1. HISTORICAL PERSPECTIVE

The history of Voronoi diagrams can be traced back to the middle of the nineteenth century. Although the spectrum of scientific disciplines that include interest in Voronoi diagrams is broad, three aspects have been emphasized:

(1) Their use in modeling natural phenomena
(2) The investigation of their mathematical, in particular, geometrical, combinatorial, and stochastic properties
(3) Their computer construction and representation

Accordingly, Voronoi diagrams are useful in three respects: As a structure per se that makes explicit natural processes, as an auxiliary structure for investigating and calculating related mathematical objects, and as a data structure for algorithmic problems that are inherently geometric. In all three applications, efficient and practical algorithms for computing Voronoi diagrams are required. Since the first application bears the initial seed for their investigation, let us consider the role that Voronoi diagrams play in the natural sciences first.

1.1 Natural Scientist's Viewpoint

To visualize the appearance of a Voronoi diagram in nature, one could think of the three-dimensional space being subdivided into a manifold of crystals: From several sites fixed in space, crystals start growing at the same rate in all directions and without pushing apart but stopping growth as they come into contact. The crystal emerging from each site in this process is the region of space closer to that site than to all others. In other words, the regions form a Voronoi diagram in three-space.

1.1.1 Domains of Action

Most of the early work on Voronoi diagrams was motivated by crystallography.

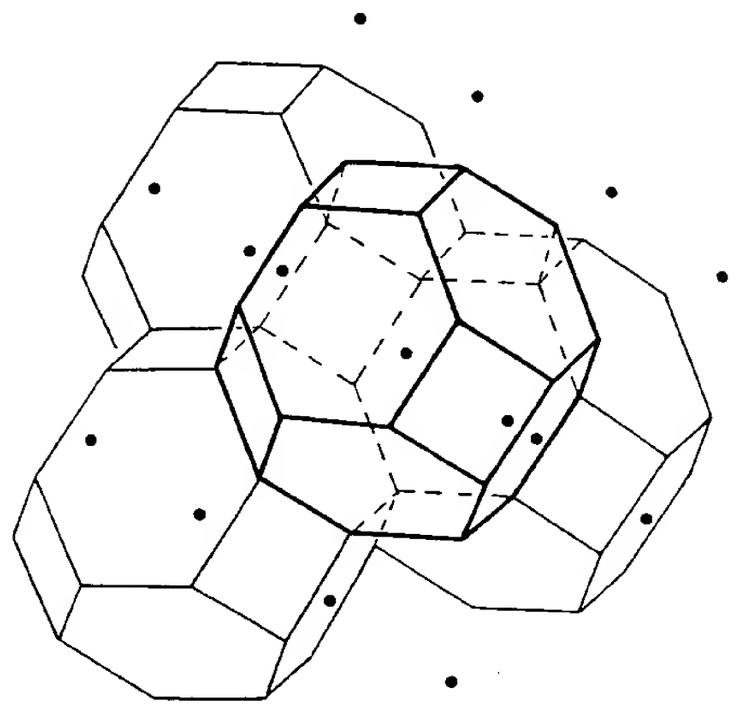


Figure 5. Cubic crystal structure.

The objective in this respect was the study of regions arising from regularly placed sites. Figure 5 illustrates some of the regions defined by 15 cubically ordered sites. In his comprehensive article on crystal structures, Niggli [1927] calls them *Wirkungsbereiche* (*domains of action*), a term widely used nowadays and used, with many synonyms, even in the 1930s as demonstrated in the clarifying note by Nowacki [1933]. Much effort has been devoted to the fundamental crystallographical question: Which types of domains of action are capable of filling the plane or three space completely if only congruent copies (and certain motions) are to be used? Significant work was done, among others, by Niggli [1927], Delaunay (Delone) [1932], Nowacki [1976], and by Koch [1975]. In fact, the question is a deep mathematical one, and we will come back to it in the discussion of mathematical aspects of Voronoi diagrams.

1.1.2 Wigner-Seitz Zones

A physicochemical system consists of a number of distinct sites, typically molecules, ions, or atoms. Equilibrium and other properties of the system depend on the spatial distribution of the

which can be conveniently repre-
by dividing the space between
according to the nearest-neighbor
The resulting Voronoi regions are
called *Wigner-Seitz zones* after
r and Seitz [1933], who were the
) use them in metallurgy. Frank
asper [1958] use these zones in the
gation of complex alloy structures;
1970] bases his survey of cubic
structures on an intimately re-
oncept. Allotting zones to sites is
rest to molecular physicists, bio-
ts, material scientists, and physi-
mists. Via this approach Brostow
cotte [1975] estimate the coordina-
umber of liquid argon, David and
[1982] study certain solvation
ires, Brumberger and Goodisman
interpret the small-angle scatter-
catalysts, and Augenbaum and
[1985] treat large-scale hydro-
ic codes—just to name a few.

ohnson-Mehl and Apollonius Model

earest-neighbor rule (or equiva-
the crystal growth model men-
earlier) forces the Voronoi regions
convex polyhedra. Allowing the
s to start their growth at different
gives rise to hyperbolically shaped
s. The resulting *Johnson-Mehl*
was proposed by Johnson and Mehl
as a more realistic model of struc-
r minerals. Figure 6 gives an il-
ion; crystals are augmented with
ate of birth.
tals growing simultaneously but
erent rates give rise to spherically
l regions forming the *Apollonius*
This structure can also be ob-
as cell structures of plants or in
made out of soap bubbles [Matzke
estler 1946; Smith 1954; Williams
It further appears as covering ar-
plants and as areas of best-
ed transmitters [Sakamoto and
i 1988]. Weaire and Rivier [1984]
comprehensive survey and various
ical data. Interestingly, the equi-
n state of a spider web constitutes
eralized, although still polygonal,
Voronoi diagram. This follows from investigations by Maxwell [1864]. Figure 7 shows sites for a spider web such that the distance between neighborhood sites corresponds to the tension of the edge they define.

1.1.4 Thiessen Polygons

Geographical interest in Voronoi diagrams originates with the climatologist Thiessen [1911], who assigned proximity polygons to observation sites in order to improve the estimation of precipitation averages over large areas. His method was worked out in detail by Horton [1917] who proposed the name *Thiessen polygons*, crediting Thiessen with the idea. As reported in Boots [1979], Thiessen polygons play four roles in geographical research: As models for spatial processes, as nonparametric techniques in point-pattern analysis, as organizing structures for displaying spatial data, and for calculating individual probabilities in point patterns. In particular, we mention Tuominen [1949] and Snyder [1962] (applications to urban planning), Gambini [1966] and Boots [1979] (market areas; generalized Thiessen polygons are used there, in particular, the Johnson-Mehl and Apollonius models), Mollison [1977] (ecological contact models), McLain [1976] (spatial interpolation), Arnold and Milne [1984] (cartography), and Okabe et al. [1988] (facility location). For surveys of Thiessen polygons from a geographical and economic point of view, see Boots [1986], Eiselt and Pederzoli [1986], and Sibson [1979].

Suzuki and Iri [1986] report on the importance of recovering the sites from a given subdivision of a geographical area. This inverse process of constructing Thiessen polygons arises in the optimal outline of school districts or of voting precincts. In geographical variation analysis, connectivity graphs for sites are a valuable tool [Matula and Sokal 1980]. Among them are the minimum spanning tree or Prim shortest connection network (Figure 8), the Delaunay triangulation, and the Gabriel graph. These three

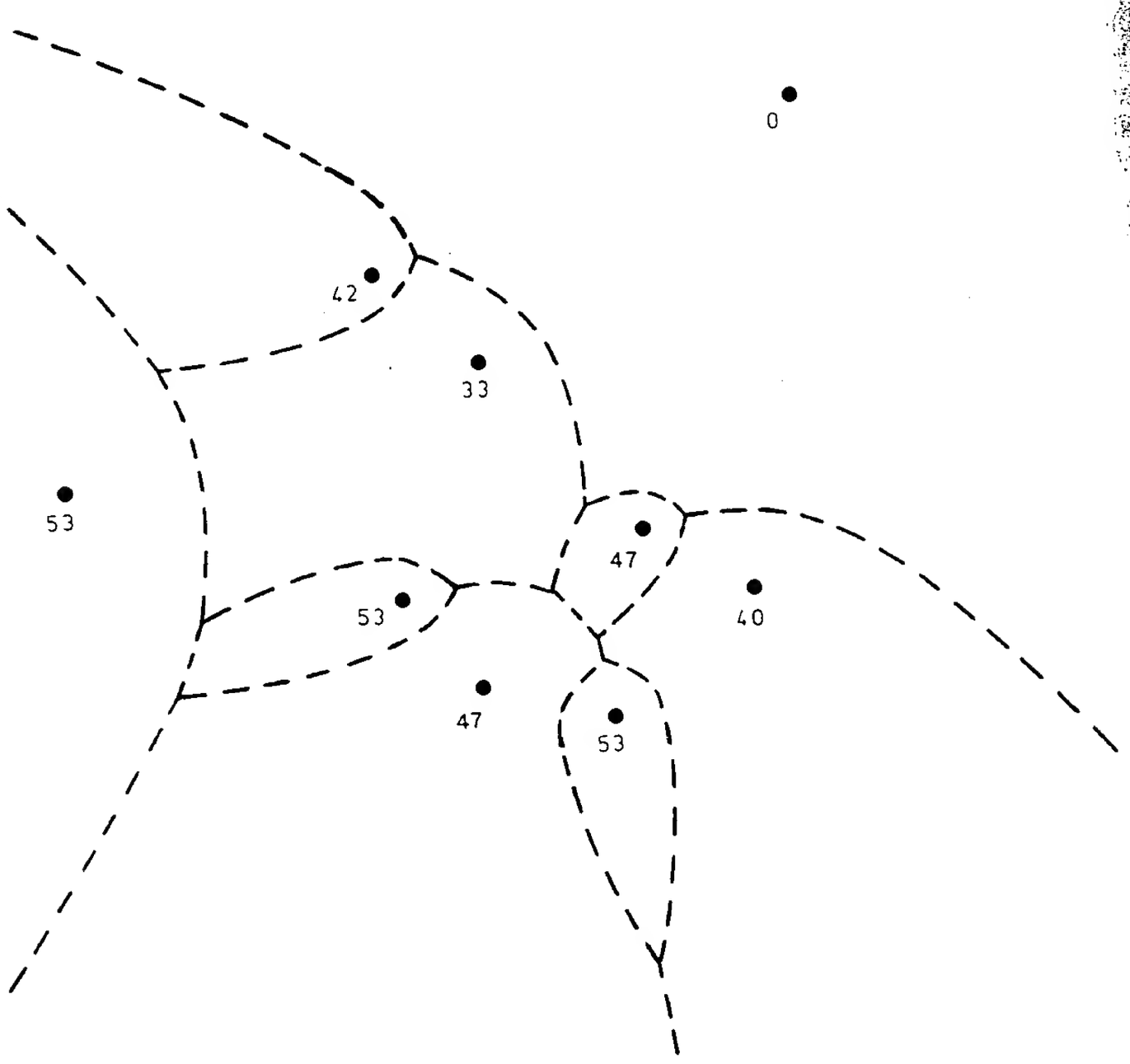


Figure 6. Johnson-Mehl model.

concepts are intimately related to Thiessen polygons as will be discussed later.

1.1.5 Blum's (Medial Axis) Transform

Yet another name for the Voronoi diagram is popular in the natural sciences: *Blum's transform* of a set of sites. Concerned with biological shape and visual science, Blum [1967] used it for modeling new descriptors of shape. In general, the main problem in pattern recognition is to extract characterizing elements from a given (site) pattern. When no geometrical model of the pattern is available, its structural description may be based on the notion of neighborhood of a site and thus, in particular, on Blum's transform. Fairfield [1979] successfully compared contours obtained in this way with perceptual boundaries studied by Gestalt psychologists. As pointed out by Blum [1973], curvature properties of a given contour correspond to topological properties of its transform (compare Figure 9). The smooth pieces of the contour (solid) should be viewed as sites of generalized shape. By definition, each point of the transform (dashed) is equidistant from at least two sites. In this context, the transform often is called the *medial axis* or *skeleton* of the contour. The medial axis of digitized contours was exploited in image processing by Philbrick [1968], Montanari [1968], and Lantuejoul and Maisonneuve [1984].

Although the foregoing list of applications of Voronoi diagrams in the natural

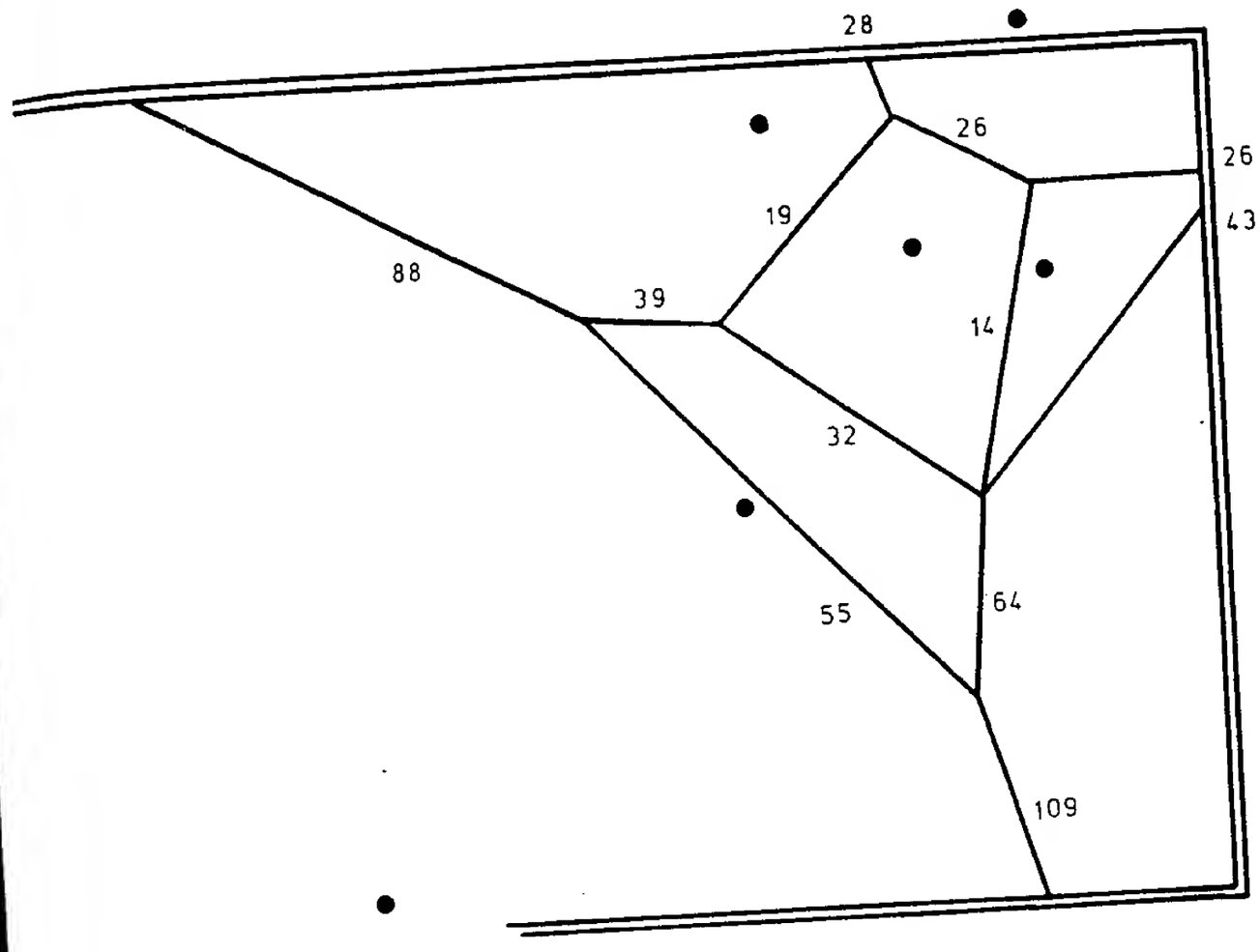
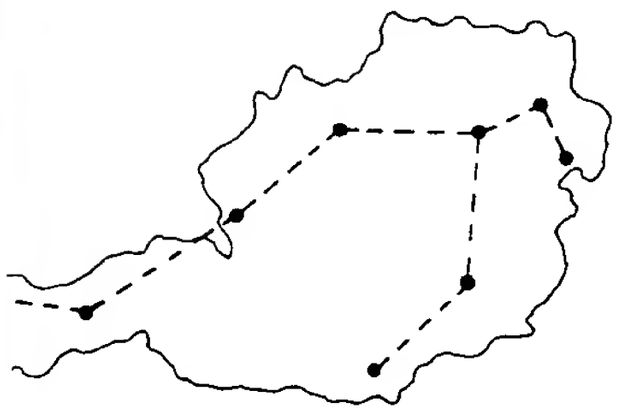


Figure 7. Spider web.



3. Minimum spanning tree for the nine pitals of Austria.

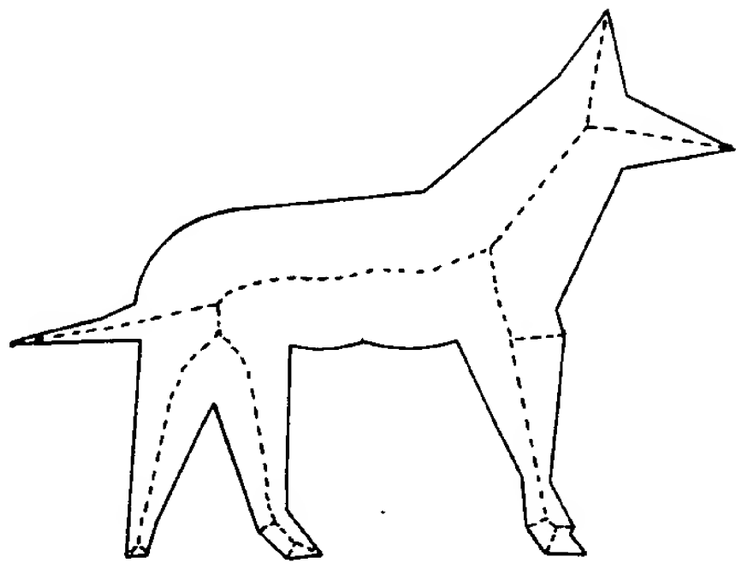


Figure 9. Contour and its medial axis [Philbrick 1968].

es is not complete, we refrain from r details and turn now to their role mathematical environment.

thematician's Viewpoint

ry earliest motivation for the study onoi diagrams stems from the the- quadratic forms. Gauss [1840] ed that quadratic forms have an retation in terms of Voronoi ms for sites that are parallelohe- ordered in space. His idea was exploited by Dirichlet [1850] to establish, among other results, a simple proof of the unique reducibility of quadratic forms. A careful generalization to higher dimensions was provided by Voronoi [1908]. To honor the pioneering work of these mathematicians, the construct has been referred to as the *Dirichlet tessela- tion* or the *Voronoi diagram*, and indeed

these terms are the most popular ones in the vast body of literature.

The following review provides basic mathematical properties of Voronoi diagrams and their generalizations (many of them will also be crucial in the understanding of their algorithmic properties and applications), as well as points out certain features of Voronoi diagrams that have potential applications but may be not familiar to computational geometry researchers.

1.2.1 Regularly Placed Sites

Mathematical interest in Voronoi diagrams can be divided according to whether regularly or irregularly placed sites are involved. A main area where Voronoi diagrams for regularly placed sites have been used for a long time is geometrical crystallography.

Let R^d denote the d-dimensional Cartesian space. A *tiling* of R^d is a covering of R^d by closed sets whose interiors are pairwise disjoint. Tilings of R^d by convex polyhedra with the property that the group of motions mapping the tiling onto itself is a crystallographic group [Bieberbach 1912] are of particular interest. Their polyhedra are necessarily congruent and were called *stereohedra* by Federoff [1885] and *fundamental domains* by Schönflies [1891]. One of the central questions of geometrical crystallography has been to enumerate all stereohedra and to classify them according to their crystallographic groups.

Most progress was made for the subclass of *plesiohedra*. They can be interpreted as Voronoi regions and were called *special fundamental domains* by Schönflies [1891] and *domains of action* by Niggli [1927]. Laves [1930] and Delaunay et al. [1978] enumerated all plesiohedra in R^2. Applying a refined classification scheme, Delaunay [1932] distinguished among 24 types of plesiohedra in R^3. A complete list of planar stereohedra was given by Grünbaum and Shephard [1987].

Delaunay [1932] showed that the number of *facets* (faces of dimension $d - 1$) of a stereohedron in R^d is finite; for instance, 390 is an upper bound for $d = 3$. So, by Tarski's [1951] decidability theorem, all spatial stereohedra can be "effectively" determined. In spite of this fact, their enumeration is still incomplete. Also, an enumeration method for plesiohedra mentioned in Delaunay [1963] does not yield a practical algorithm. Among others, Nowacki [1976], Koch [1973], and Engel [1981] discovered spatial plesiohedra with 18, 23, and 38 facets, respectively. Figure 10 depicts a plesiohedron with many facets. It seems surprising that its congruent copies are capable of filling three-space completely.

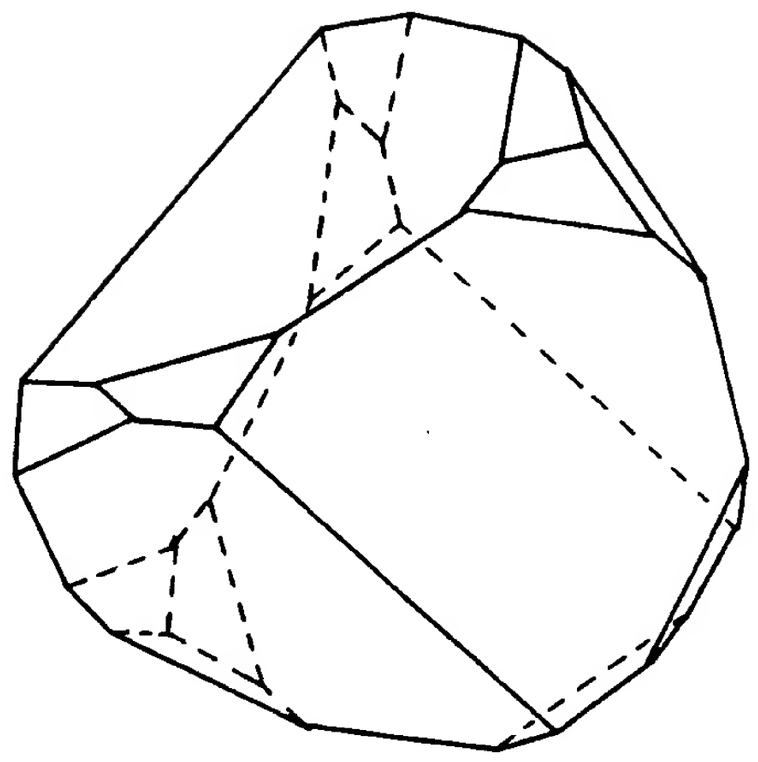


Figure 10. Plesiohedron with 18 facets [Grünbaum and Shephard 1980].

The methods used to obtain plesiohedra with large numbers of facets are probably as interesting as the numbers themselves. After carefully choosing a set of regularly placed sites (dependent of several variable parameters), certain parts of its Voronoi diagram are constructed via computer [Engel 1981]. For excellent sources on tilings, refer to Grünbaum and Shephard [1980, 1987].

It is worthwhile to mention that Voronoi diagrams for regularly placed sites also find other applications in mathematics. They apply to numerical integration [Babenko 1977], to packing

overing problems for congruent
s [Rogers 1964], and to statistical
gations of lattice systems [Besag
Mollison 1977; Conway and Sloane

regularly Placed Sites

now turn our attention to Voronoi
ms arising from sets of irregularly
sites. A tiling of R^d by polyhedra
d a *cell complex* in R^d if the tiling
t to facet, that is, each facet of a
dron is also a facet of some other
dron in that tiling.[2] Every tiling
by plesiohedra is a cell complex.
oronoi diagram for irregularly
sites is still a cell complex
gh its regions are no longer
ent polyhedra.

fact gives rise to various ques-
Motivated by the problem of find-
ensest sphere packings, Rogers
posed the following extremal prob-
low big is the smallest possible
i region with t facets and defined
s with minimum distance two? Es-
s of this and related quantities
erived by Muder [1988a, 1988b] for
regions.

h work on general Voronoi dia-
is concerned with their combinato-
operties and, in particular, with
ize, that is, their numbers of faces
ous dimensions as a function of the
r n of sites considered. The size of
noi diagram is an important quan-
ice it relates the amount of space
to store this structure to the in-
ze. Answering Crum's problem,
ey and Vranch [1977] exhibit an
rily large set of Voronoi polyhedra
each pair of which shares a facet.
ample they give shows that the
a Voronoi diagram in R^3 is $O(n^2)$;
as also observed by Preparata
As already mentioned, a Voronoi
m in the plane has size $O(n)$ since
ges and vertices form a planar

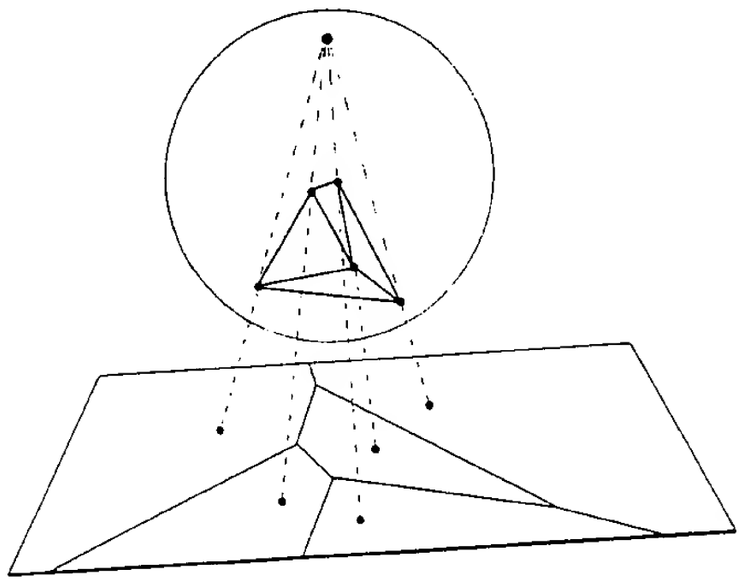


Figure 11. Voronoi diagrams are related to convex hulls.

graph. The size of higher dimensional diagrams has only recently been analyzed completely. Klee [1980] and Brown [1980] showed that Voronoi diagrams in R^d are equivalent to certain convex *polyhedral surfaces* in R^{d+1} in a strong sense. Hence, known results on the size of polyhedra carry over, in particular, the so-called upper and lower bound theorems [Brondsted 1983]. Exploiting this relationship, exact bounds on the numbers of individual faces of d-dimensional Voronoi diagrams were derived by Seidel [1982] and by Paschinger [1982]. Figure 11 illustrates Brown's original transform that relates the Voronoi diagram in R^2 to a convex hull in R^3 via stereographical projection.

1.2.3 Generalized Voronoi Diagrams

Interestingly, not every convex polyhedral surface in R^{d+1} is related to a Voronoi diagram in R^d. For a more general class, a one-to-one correspondence can be established. Paschinger [1982] and Aurenhammer [1987a] showed that *power diagrams* are equivalent to the boundary projection of convex polyhedral surfaces.[3] This generalization assigns a particular weight w to each of the given sites p and replaces the euclidean distance $\delta(x, p)$ between a point x and p by

, a tiling if facet to facet if and only if it is
ace for faces of any dimensions less than d
and Ryskov 1987].

[3] Such projections are commonly called *Schlegel diagrams* in discrete geometry [Grünbaum 1967].

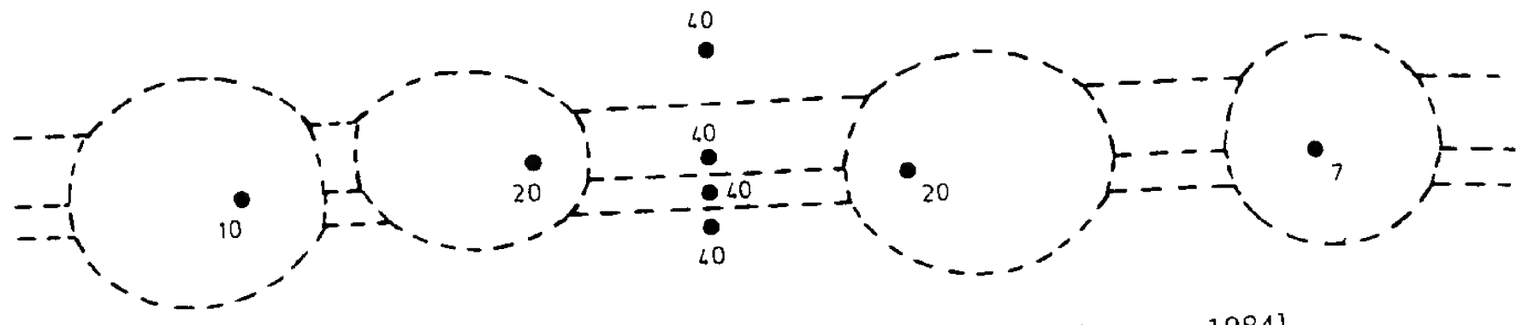


Figure 12. Apollonius model [Aurenhammer and Edelsbrunner 1984].

$\delta^2(x, p) - w$. Intuitively speaking, w expresses the capability of p to influence its neighborhood. In particular, the pair (p, w) may be interpreted as a sphere with center p and radius $\sqrt{w}$ if $w > 0$. As an important property, the separator defined by two weighted sites in R^2 is still a straight line: the power line of two circles. Figure 4 shows a power diagram in the plane. Power diagrams already appeared in Dirichlet [1850]—therefore, often being called *generalized Dirichlet cell complexes*—and continued to be objects of interest, especially in sphere packing [Rogers 1964], illuminating spheres [Linhart 1981], and the geometry of numbers [Gruber and Lekkerkerker 1988].

The concept of weighting the sites gives rise to several other useful types of diagrams. Weighting the euclidean distance by a *multiplicative* constant yields the Apollonius model, investigated by Aurenhammer and Edelsbrunner [1984]. The separator of two sites in the plane describes their Apollonius circle. From Figure 12 it can be seen that regions may be disconnected and may partition the plane into $\Theta(n^2)$ connected components. *Additive* weights give rise to the Johnson-Mehl model (compare Figure 6). For many additional types, see Ash and Bolker [1986].

At this point, let us modify the concept of Voronoi diagram in another way. The *order-k Voronoi diagram* of n sites is a partition of R^d into regions such that any point within a fixed region has the same k closest sites. Its regions are convex polyhedra that form a cell complex in R^d. For $k = 1$, the classical Voronoi diagram is obtained. In the case of $k = n - 1$, the structure is often referred to as the *furthest site Voronoi diagram* since now the region of a site contains all points in space furthest from it. Figure 13 shows an order-2 Voronoi diagram for 8 sites in the plane. Note that only 15 of the 28 pairs of sites define a region and that a separator may yield more than one edge. (An example of a furthest site Voronoi diagram is given in Figure 24.)

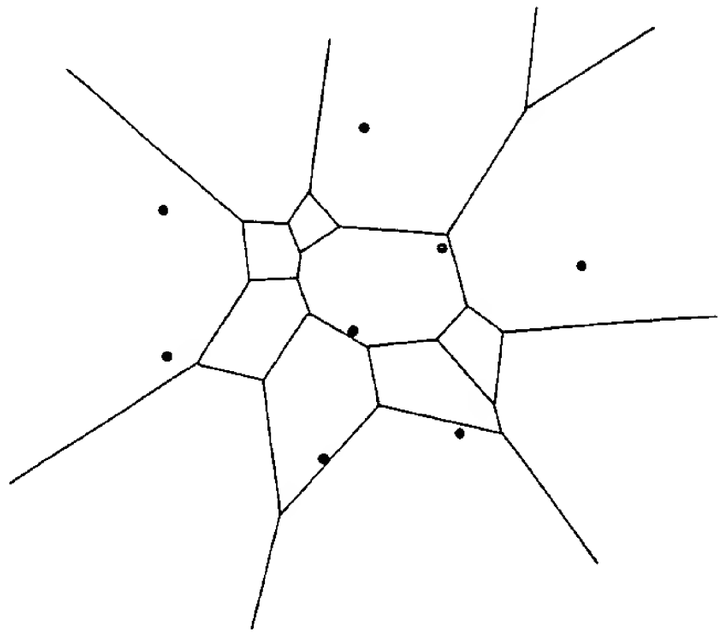


Figure 13. Planar order-2 Voronoi diagram [Shamos 1978].

Early treatments of order-k diagrams are found in Miles [1970] and in Shamos and Hoey [1975]. Lee [1982a] succeeded in proving exact upper bounds on the maximum numbers of regions, edges, and vertices of a planar order-k Voronoi diagram. Asymptotically these numbers are $\Theta(k(n - k))$. The fascinating relationship between Voronoi diagrams, weighted diagrams, order-k diagrams, and other geometric objects was worked out by Edelsbrunner et al. [1986], Aurenhammer [1987a], and

hammer and Imai [1988]. Within ontext, the concepts of power dia- and *hyperplane arrangement* play ral role. Section 3.1 is devoted to a ed description of this material. In ular, various results on the size and computational complexity of con- ing such diagrams are obtained. ver, still little is known on the size order-k Voronoi diagram in higher sions; consult [Edelsbrunner and 1986]. Only for the furthest-site m exact upper bounds are avail- Seidel 1987]. The number of re- of an order-k Voronoi (or power) m is closely related to the num- *k-sets* of a finite point set. Deter- g these numbers belongs to the open problems in combinatorial try.

e the usefulness of known types of oi diagrams had been realized, fur- generalizations were attempted. ation mostly came from applica- in computational geometry. In or- meet practical needs, *general ce functions* like the L_p-metrics 1980a] or convex metrics [Chew rysdale 1985] were taken to define onoi diagram. Also, the *shape of* as varied while using a cannonical ion of the euclidean distance func- line segments [Kirkpatrick 1979], segments [Yap 1987], and disks or l convex sites [Leven and Sharir 1987] have been considered. If the nts form a closed curve, their dia- is just the medial axis of a contour are Figure 9). Disks can be viewed ters weighted additively by their thus giving rise to the Johnson- model. This list is not meant to be stive; we shall be concerned with matical and computational proper- of varous generalized Voronoi ms in later sections.

us mention one more way of gener- on: *changing the underlying space.* h and Im Hof [1979] investigated ehavior of Voronoi regions in nn manifolds. Brown [1980], nger [1982], and Yap [1987] ob- that Voronoi diagrams on the sphere and on the torus, respectively, are closely related to their equivalents in the euclidean space of the same dimension. Diagrams on three-dimensional polyhedral surfaces and on the three-dimensional cone have been treated by Mount [1985] and by Dehne and Klein [1987], respectively.

1.2.4 Delaunay Triangulations

Hand in hand with the investigation of Voronoi diagrams goes the investigation of related constructs. Among them, the *Delaunay triangulation* is most prominent. It contains a (straight-line) edge connecting two sites in the plane if and only if their Voronoi regions share a common edge. The structure was introduced by Voronoi [1908] for sites that form a lattice and was extended by Delaunay [1934] to irregularly placed sites by means of the empty-circle method: Consider all triangles formed by the sites such that the circumcircle of each triangle is empty of other sites. The set of edges of these triangles gives the Delaunay triangulation of the sites.

The planar Voronoi diagram and the Delaunay triangulation are duals in a graphtheoretical sense. Voronoi vertices correspond to Delaunay triangles, Voronoi regions correspond to sites, and edges of both types correspond by definition. From Figure 14 it can be seen that Delaunay edges (solid) are orthogonal to their corresponding Voronoi edges (dashed)—but do not necessarily intersect them—and that the boundary of the convex hull of the sites consists of Delaunay edges. The duality immediately implies upper bounds of $3n - 6$ and of $2n - 4$ on the number of Delaunay edges and triangles, respectively (compare Introduction). The Delaunay triangulation and its duality to Voronoi diagrams generalize to higher dimensions in an obvious way. By results by Dewdney and Vranch [1977], the Delaunay triangulation in R^3 may already be the complete graph on n sites, thus having $\binom{n}{2}$ edges. For a

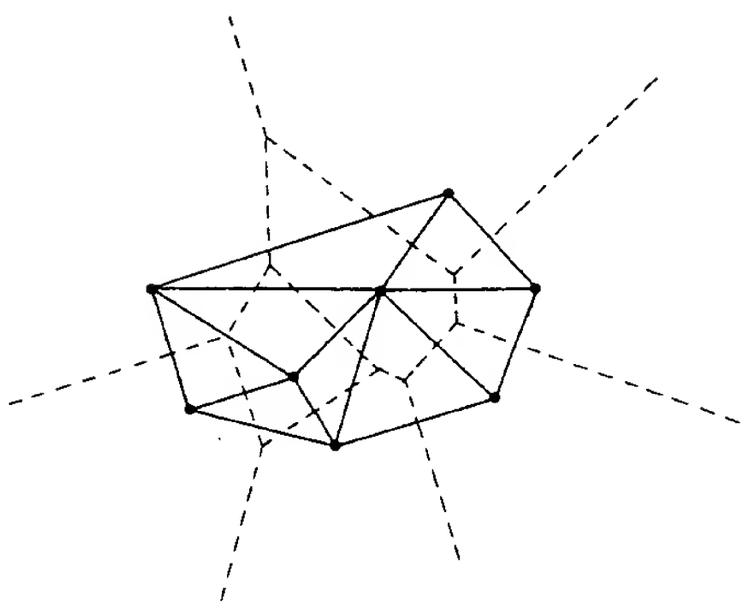


Figure 14. Voronoi diagram and Delaunay triangulation are duals.

catalog of properties of higher dimensional Delaunay triangulations arising from site lattices see Gruber and Lekkerkerker [1988].

Several interesting properties are known for the Delaunay triangulation in the plane. It was first observed by Sibson [1977] that this triangulation is *locally equiangular*. This property holds when, for any two triangles whose union is a convex quadrilateral, the replacement of their common edge by the alternative diagonal does not increase the minimum of the six interior angles concerned. Actually, the Delaunay triangulation is the only one with this property that particularly shows its uniqueness. For sites being in general position (the Delaunay triangulation may contain more-sided faces if four sites are cocircular), Edelsbrunner [1987] showed that local equiangularity is equivalent to *global equiangularity*. The triangles define the lexicographically largest list of sorted angles. A similar result holds if only the smallest angle of each triangle is considered [Lawson 1977]. Note that the Delaunay triangulation thus maximizes the minimum angle over all triangulations of a given set of sites. On the other hand, a simple example shows that the maximum angle is not minimized. It is interesting to note that, by the empty circle property, any triangulation without obtuse angles must be Delaunay. Triangulations without "extreme" angles are desirable in finite element and interpolation methods.

Lawson [1972] gave a counterexample to Shamos and Hoey's [1975] conjecture that the Delaunay triangulation has minimum total edge length. It does not even approximate the shortest triangulation [Manacher and Zobrist 1979], and in fact it may be as long as any triangulation [Kirkpatrick 1980]. It is, however, close to optimal on the average [Lingas 1986a]. A different criterion of optimality is mentioned in McLain [1976]. For each triangle, all its points should be at least as close to one of its defining sites as to any other site. This property is not shared by the Delaunay triangulation, as is claimed there.

As an important fact, the Delaunay triangulation is a supergraph of several well-known and widely used graphs spanned by a set of sites in the plane. Among them are the *minimum spanning tree* (or *Prim shortest connection network*) introduced by Kruskal [1956] and Prim [1957], the *Gabriel graph* introduced by Gabriel and Sokal [1969], and the *relative neighborhood graph* introduced by Toussaint [1980]. Section 2.4 gives definitions of these graphs and describes their interrelations.

The Delaunay triangulation can be exploited to find certain *linear combinations* among its defining sites. Each site not on the convex hull can be represented as a weighted mass center of the sites adjacent in the triangulation; see Sibson [1980] who mentions applications to surface smoothing. Aurenhammer [1988b] generalized this result to power diagrams and their order-k modifications. In particular, *Gale transforms* of the sites may be derived in this way [Aurenhammer 1990b]. Gale transforms are a versatile tool in the investigation of high-dimensional convex polyhedra [Grünbaum 1967].

The generalized Delaunay triangulation obtained from power diagrams gains in importance from the following recognition problem: Given some cell complex, can it be interpreted as a Voronoi diagram? A cell complex in R^d may be

l as a power diagram if and only if
ll complex admits a certain dual
uct, called the *reciprocal figure* by
ell [1864], Crapo [1979], Whiteley
, and Ash and Bolker [1986]. The
ocal figure is completely character-
y the properties of the Delaunay
ulation for power diagrams. Using
ocal figures as a criterion,
hammer [1987b] showed that sev-
well-known types of cell com-
are power diagrams. Moreover,
oi diagrams can be recognized
heir defining sites can be re-
in an efficient way using recip-
figures [Aurenhammer 1987c]. For
e survey on reciprocal figures,
h et al. [1988]. Essentially distinct
a for cell complexes to be classical
be multiplicatively or additively
ted, Voronoi diagrams were pro-
by Ash and Bolker [1985, 1986].
[1967] first suggested that gener-
haped sites could be reconstructed
the shape of the regions they de-
Calabi and Hartnett [1968] elabo-
on this question in some detail.
nition problems of this kind find
ations, aside from geography and
nics, in statics and in computer
e.

s list of geometric and combina-
properties of Delaunay triangu-
s is not complete. We shall
arious others while discussing
ations in computational geometry.

Stochastic Properties

, we have surveyed a good deal of
ch on geometric and combinatorial
s of Voronoi diagrams. Another
icant stream of investigations con-
the determination of statistical data
the diagrams obtained from ran-
distributions of sites. One of the
st and strongest motivations for
ing stochastic properties of Voronoi
ams stems from their practical rele-
to physical and chemical processes,
ally in metallurgy and crystallog-
[Johnson and Mehl 1939]. Accord-
most efforts concentrated on sites
buted in R^2 and in R^3. In his valu-
able paper, Meijering [1953] derived means for the volume, the total boundary area, and the total edge length of the regions of a Voronoi diagram in R^3 arising from a *Poisson field* of sites, as well as the average number of vertices, edges, and facets. Several of these quantities are also given for the Johnson-Mehl model. Further progress was made by Gilbert [1961] who determined the variances for the volumes of such Voronoi regions. He also found some variances associated with plane or line section through regions. Based on experimental results, Kiang [1966] gave a formula for the random size distribution of Voronoi regions in R^1, along with a conjecture for its generalization to R^2 and R^3. The conjecture is, however, incorrect according to results in Gilbert [1961]. Figure 15 shows his obtained density functions in R^1 (solid), in R^2 (dashed), and in R^3 (dotted). Observed volume over mean volume is scaled.

Miles [1970] made a thorough study of Voronoi diagrams induced by a planar *homogenous site process*. Expectations of the edge number, edge length, perimeter, and area of the regions are given. Further results concern the Delaunay triangulation and the order-k family of diagrams. Relevant experimental data can be found in Crain [1978]. His observed frequencies of edges per polygon are of particular interest since no theoretical results are presently available. An important result by Dwyer [1989] shows that the expected number of vertices of the classical (closest or furthest site) Voronoi diagram in d dimensions is only $O(n)$ if the n sites are uniformly drawn in a hyperball. At this place, we mention only marginally that several results have been obtained that concern the behavior of Voronoi diagrams for sites being introduced in *random order* rather than being drawn by some probability distribution. We shall see such results in Section 1.3.

Space constraints preclude mentioning all related work. To name a few authors, we refer to Newman et al. [1983] (number of nearest neighbors in d dimensions), Weaire and Rivier [1984]

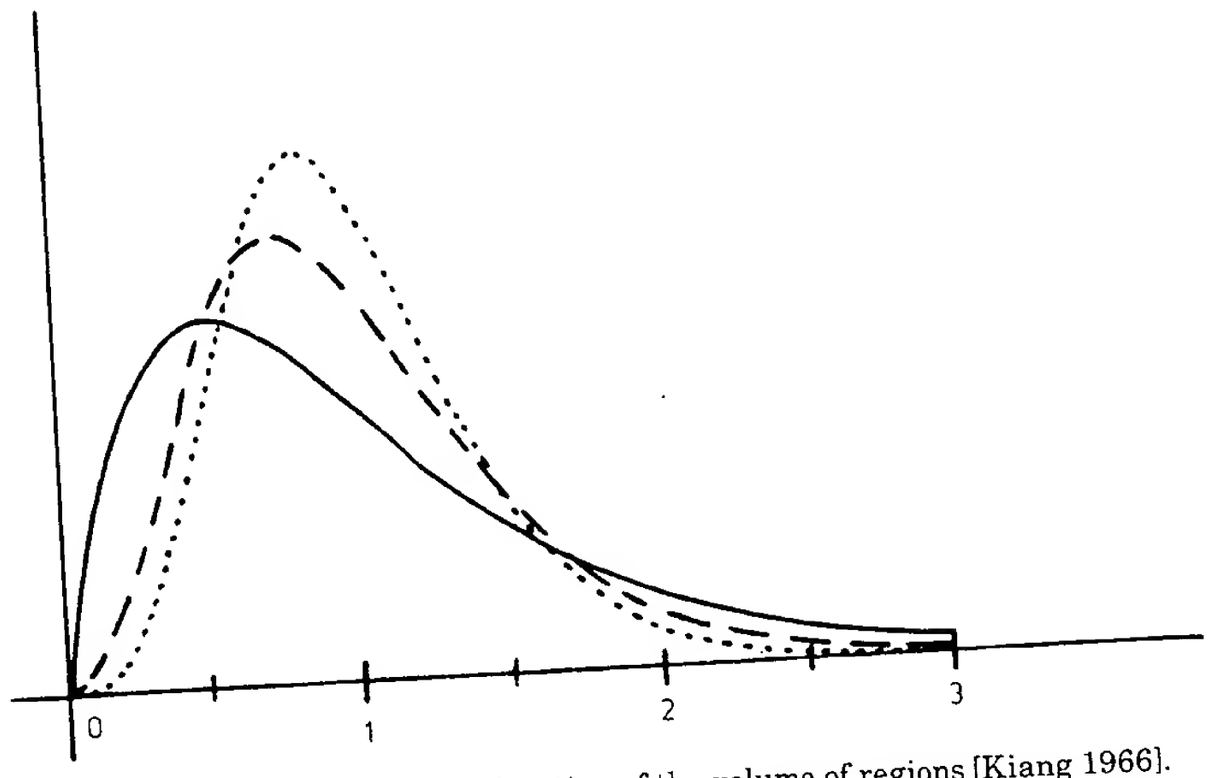


Figure 15. Empirical density function of the volume of regions [Kiang 1966].

(generalized diagrams), Besag [1974], Mollison [1977], Cruz Orive [1979], Conway and Sloane [1982] (site lattices; applications to spread of epidemics and to coding theory are given), and Aurenhammer et al. [1991] (probabilistic distance functions). The reader interested in geometrical probability in general and stochastic properties of cell complexes in particular may consult the expository papers by Moran [1966, 1969], Little [1974], and Baddeley [1977].

1.3 Computer Scientist's Viewpoint

We have documented the remarkable role that Voronoi diagrams play in the mathematical and applied natural sciences. Yet for a long time their practical usefulness suffered from the absence of reasonably simple and efficient methods for their computation. This section reviews methods for the computer construction and representation of Voronoi diagrams. Algorithmic applications of Voronoi diagrams and of related structures are discussed in Section 2.

1.3.1 Early Algorithms

Since Voronoi diagrams have been used for decades by natural scientists, many intuitive construction rules were proposed. The earliest diagrams were drawn with pencil and ruler; see Horton [1917] or Kopec [1963] who mentions problems of ambiguity if many sites lie on a common circle. Probably the most obvious approach is to delineate the diagram in the plane region by region, by singling out those separators that contribute to edges of the current region. A provisional and admittedly inefficient version of such an algorithm was described by Rhynsburger [1973]. Other early algorithms build up the diagram vertex by vertex [Brassel and Reif 1979] or by *incremental insertion* of sites (i.e., of their regions) [Green and Sibson 1977].

Figure 16 illustrates the insertion of a site p that involves two tasks. First, we need to find the current region in which p falls. Let q be the site defining this region; the separator of p and q then will contribute an edge, e, to p's region. Second, the boundary of p's region (bold) is created edge by edge, starting with e. During this process, the parts of the old diagram closest to p (dashed) are traversed and deleted. These parts are specified by lying on p's side of the separators of p and its new neighbors. If appropriately implemented, the second task requires time proportional to

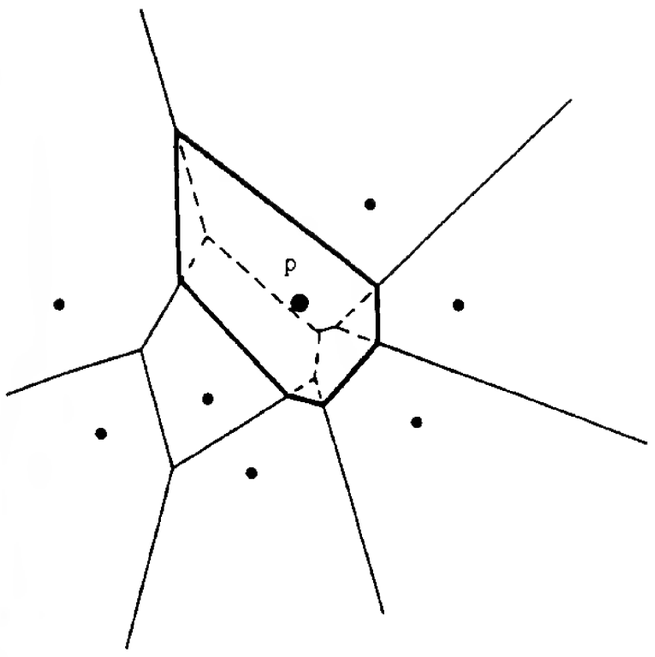


Figure 16. Inserting a Voronoi region.

iumber of edges deleted. This num-
s $O(i)$ in the worst case (i denotes
iumber of sites inserted so far) since
ave to delete a planar graph with at
i regions. The first task requires
time in the worst case, but a simple
istic will achieve $O(\sqrt{i})$ expected
. Experimental results showed that
method is quite efficient and thus is
cable to rather large sets of sites. As
the other algorithms mentioned
e, however, the worst-case behavior
n^2) for n given sites.
veral intuitive methods are also
able for the Voronoi diagram in R^3.
compute the induced cell complex
by facet [Brostow et al. 1978], ver-
›y vertex [Finney 1979], or via its
, the Delaunay triangulation
emura et al. 1983]. Potentially, they
ire $O(n^4)$ time in the worst case,
›ugh they may perform quite well for
›us distributions of sites. For $d \geq 4$
nsions, insertion strategies working
tly [Bowyer 1981] or based on the
unay triangulation [Watson 1981]
been used to construct a suitable
›inatorial representation of the
noi diagram. Their time complexity
$n^{1+1/d}$) and $O(n^{2-1/d})$, respectively,
ided the sites are "well distributed."
should be contrasted, however,
ıst the maximal size of a Voronoi
diagram in R^d that grows exponentially with d. A three-dimensional implementation of Watson's algorithm is discussed in Field [1986]. Avis and Bhattacharya [1983] propose an $O(n^{\lceil d/2 \rceil + 1})$ time algorithm for determining all vertices of the d-dimensional diagram. They also outline a linear programming method for calculating the d-dimensional Delaunay triangulation but give no concrete complexity analysis.

Only recently have optimal or near-optimal algorithms for constructing Voronoi diagrams been devised. The reason is that only in the last few years have powerful algorithmic techniques been fully developed and exploited for computational geometry purposes.

1.3.2 Speeding Up Insertion

The process of building a Voronoi diagram in the plane by incremental insertion of sites stands out by its simplicity. Originally having an $O(n\sqrt{n})$ expected performance, the Green-Sibson algorithm may be polished up to run faster for several distributions of the sites. By introducing suitable orderings of the sites, the expected time for finding the region the next site falls in and for integrating the new region can be lowered to $O(1)$ [Ohya et al. 1984a, 1984b].[4] This gives an expected runtime of $O(n)$. Clearly this is the best we can hope for because the size of the diagram is a trivial lower bound for the time needed to compute it.

An alternative approach to speeding up insertion is *randomization*. Based on a general result by Clarkson and Shor [1988], Mehlhorn et al. [1990] showed that inserting the sites in random order yields an $O(n \log n)$-time performance with high probability. This complexity is independent of the distribution of the sites; expectation is taken over all possible permutations of the sites. The algorithm extends to the class called abstract

[4] Recall in this context that the average number of edges of a region is less than six.

Voronoi diagrams by Klein [1989] without increase of runtime. This general concept includes power diagrams and diagrams defined by line segments or by L_p-metrics. Guibas et al. [1990] propose an even more practical version of randomized incremental construction of the classical Voronoi diagram.

As with other geometrical algorithms, the problem of numerical errors arises in the construction of Voronoi diagrams. For example, sites nearly lying on a common circle define vertices that tend to approach arbitrarily close. A proposal for making the strategy of insertion robust against numerical errors is outlined in Sugihara and Iri [1988].

The process of inserting Voronoi regions extends nicely to R^3. Regions are convex polyhedra that can be constructed facet by facet by intersecting existing regions with separators that are planes in this case. A region cannot have more than $n - 1$ facets (one for each different site) and thus by Euler's relation, has $O(n)$ edges and vertices. Therefore, one needs $O(n + t)$ time per region if t facets, edges, or vertices are deleted during its insertion; see Aurenhammer and Edelsbrunner [1984] who treat insertions in a more general cell complex in R^3. Since each component deleted has to be constructed first, an $O(n^2)$-time algorithm results. This is worst-case optimal since a Voronoi diagram in R^3 may have a size of $\Theta(n^2)$.

1.3.3 *Divide and Conquer*

A widely used method to design fast algorithms is *divide and conquer*. Shamos and Hoey [1975] observed that this method applies well to problems in computational geometry and, in particular, to the construction of a Voronoi diagram in the plane: The given set of n sites is divided into two subsets by a vertical line. The diagrams for these subsets are computed recursively and are then "merged" in the conquer step to form the total diagram. Figure 17 shows the chain of edges (bold) to be constructed during the process of merging the diagram for left sites (dashed) with the diagram for right sites (dotted).

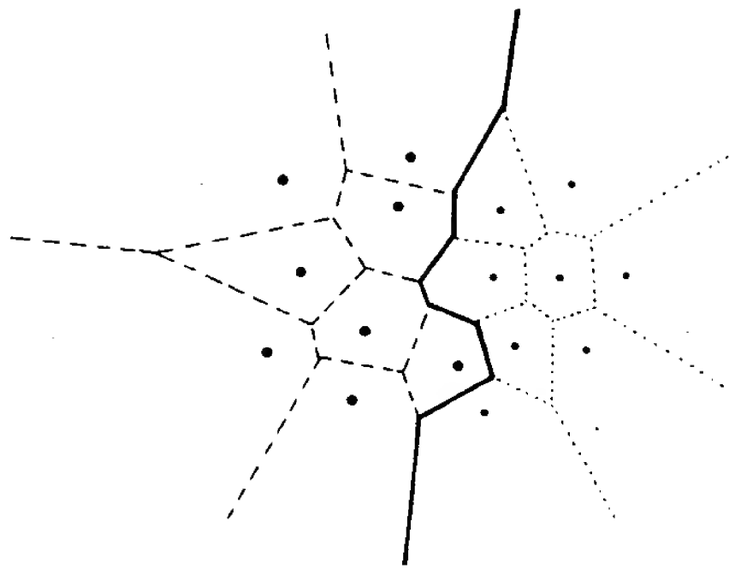


Figure 17. Merging two Voronoi diagrams [Shamos 1978].

Section 3.2 is devoted to a detailed description of the method and its generalizations. In particular, two diagrams can be merged in $O(n)$ time, which implies an $O(n \log n)$ time algorithm if the divide step is carried out in a balanced way. This is worst-case optimal since any algorithm that constructs an explicit description of a Voronoi diagram may be used to sort: Interpret an input sequence of n real numbers as a set of sites on the x-axis and construct their Voronoi diagram. Regions of sites for consecutive numbers will share an edge, so the sorted sequence can be obtained in $O(n)$ time by scanning through the regions.[5]

Although theoretically fast and elegant, the divide-and-conquer approach has certain disadvantages. Implementation details are somewhat complicated, and numerical errors are likely by

[5] Voronoi diagrams do not help for sorting general sets of sites. Seidel [1985] proved that sorting n sites in the plane with respect to their x-coordinates takes $\Omega(n \log n)$ worst-case time even when their Voronoi diagram is part of the input. It is unclear whether sorting helps for Voronoi diagrams. Fortune [1988], however, pointed out that presorting the sites in two different directions lowers the additional time for constructing the diagram in the L_1-metric to $O(n \log \log n)$.

ruction: Dividing the plane into w slabs forces the vertices defined es within a slab to approach infin- Moreover, the expected behavior is $O(n \log n)$. An $O(n)$-expected-time ithm that applies divide and con- to a certain subset of sites was ied by Bentley et al. [1980].

ide and conquer is the basis of a class of algorithms for computing alized Voronoi diagrams in the . Let us briefly list some of them in to give credit to the authors who lered them first. Shamos and Hoey] reported that their algorithm also s to the furthest site Voronoi dia- in the plane. Hwang [1979], Lee Vong [1980], and Lee [1980] consid- point sites under the L_p-metrics, as even more general distance ons were treated by Widmeier et 87], Chew and Drysdale [1985], and and Wood [1988]. Imai et al. [1985] d sites under the Laguerre dis- that is, their power diagram. rning the divide-and-conquer con- ion of Voronoi diagrams for sites general than points, we refer to atrick [1979] (line segments), Lee Drysdale [1981] (line segments or tersecting discs), Sharir [1985] rary disks), Leven and Sharir [1987] r convex bodies), Yap [1987] (curve nts), and Lee [1982b] (medial axis imple polygon). It should be noted most of the algorithms just men- l are worst-case optimal.

Higher Dimensional Embedding

forming geometrical problems into easily understood and solved ones an important role in computational try (e.g., see [Edelsbrunner 1987]). [1979] first perceived the possibil- transforming Voronoi diagrams in o convex hulls in R^3: The sites are d, via stereographical projection, oints lying on a sphere (compare e 11). This takes $O(n)$ time. onvex hull of the resulting three- sional point set is computed using lgorithm by Preparata and Hong [1977] that requires $O(n \log n)$ time. The convex hull is dual to the Voronoi diagram of the given sites in a geometrical sense. Hence $O(n)$ additional time suffices for deriving the latter. This elegant approach matches the optimal $O(n \log n)$ time bound, although it suffers from the problem of processing three-dimensional objects.[6]

An important feature of the embedding method is its easy generalization to higher dimensions. Methods for determining higher dimensional convex hulls are well established [Seidel 1981, 1986], hence efficient worst-case algorithms for computing the d-dimensional Voronoi diagram become available. Their runtime, however, increases exponentially with d according to the maximal size of a diagram. This should be contrasted with a recent result by Dwyer [1989], showing that $O(n)$ expected time suffices for computing the Voronoi diagram in constant dimensions d for uniformly distributed sites.

Brown's idea was further developed by Edelsbrunner et al. [1986] for constructing the order-k Voronoi diagram family and by Aurenhammer [1987a] for constructing the power diagram and its order-k family in general dimensions. See also Edelsbrunner [1986], Aggarwal et al. [1989a], and Aurenhammer [1990a] where order-k diagrams are obtained in different ways from their higher dimensional embeddings. We refrain from any details here and refer to Section 3.1 for a comprehensive discussion of this material. For different approaches to the computation of order-k diagrams in the plane consult Lee [1982a] and Chazelle and Edelsbrunner [1987]. Clarkson [1987] speeds up a combined version of these approaches using random sampling, achieving $O(kn^{1+\epsilon})$ expected time and

[6] In fact, since Preparata and Hong construct convex hulls by divide and conquer, this is just a three-dimensional translation of a divide-and-conquer construction. By Brown's result, simple convex hull algorithms [Clarkson and Shor's 1988] lead to simple Voronoi diagram algorithms, however.

space (for any $\epsilon > 0$) that is nearly optimal.

Embedding in higher dimensions yields algorithms for various different types of diagrams. See Aurenhammer and Edelsbrunner [1984] for the multiplicatively weighted Voronoi diagram, Aggarwal et al. [1989a] for the medial axis of a convex polygon, Edelsbrunner et al. [1989] for cluster Voronoi diagrams, and Aurenhammer and Stöckl [1988] for the peeper's Voronoi diagram.

1.3.5 *Plane-Sweep Construction*

Another powerful technique in computational geometry is the *plane-sweep technique* [Preparata and Shamos 1985]. In contrast to the embedding method, this technique decreases the dimension of the problem. Intuitively speaking, the static problem of computing a Voronoi diagram in the plane is reduced to the dynamic problem of maintaining the cross section of the diagram with a straight line. The algorithm simulates sweeping a line across the plane from below. At any point in time, the portion of the diagram below the sweep line is complete (Figure 18). Fortune [1985, 1987] first observed that updates on the sweep line can be implemented to cost $O(\log n)$ time if a certain continuous deformation of the diagram is treated. From this deformation, the original diagram can be constructed in $O(n)$ time. The method, its application, and its modification for generalized Voronoi diagrams are described in detail in Section 3.3.

The plane-sweep approach to Voronoi diagrams combines simplicity and efficiency. It achieves the optimal $O(n \log n)$ time and $O(n)$ space bounds for the classical type, the additively weighted type, and the diagram for line segments [Fortune 1987]. The method has been successfully applied by Seidel [1988] to Voronoi diagrams with line segment obstacles joining sites (and their constrained Delaunay triangulations). This type had been previously attacked by means of rather involved divide-and-conquer algorithms [Lee and Lin 1986; Lingas 1986b; Chew 1989a; and Wang and Schubert 1987]. As was reported by Rosenberger [1988], plane sweep also performs well for planar order-k diagrams even when the sites are weighted additively.

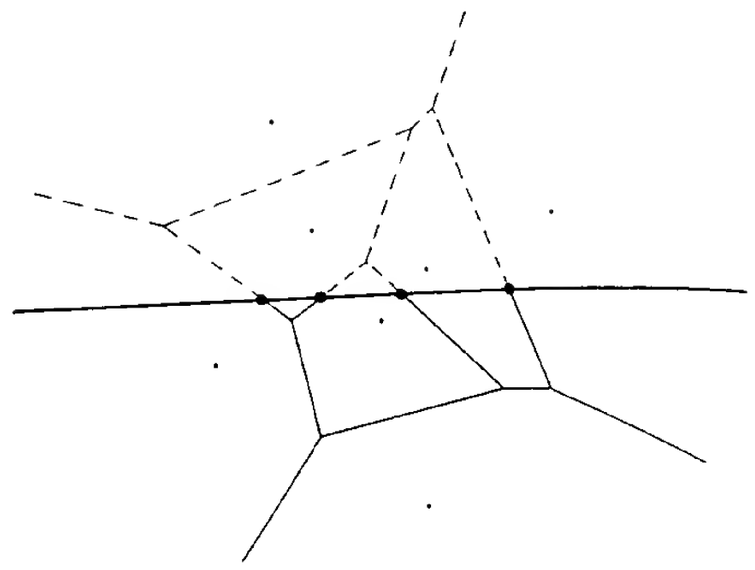
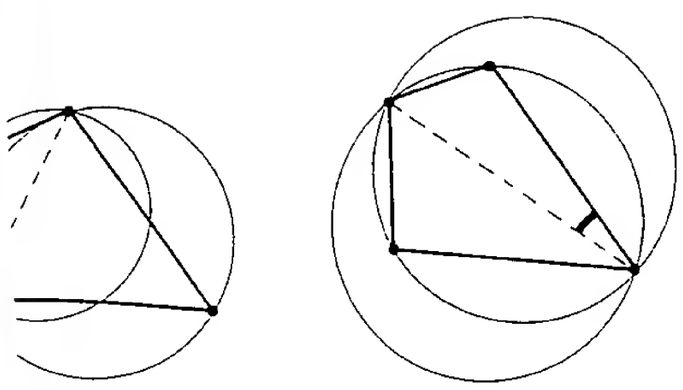


Figure 18. Maintaining the intersection with a line.

1.3.6 *Delaunay Triangulation Algorithms*

Since the Voronoi diagram and the Delaunay triangulation are duals, the combinatorial structure of either structure is completely determined from its dual. Consequently, the Delaunay triangulation of n sites in the plane can be obtained in $O(n)$ time after an $O(n \log n)$-time precomputation of the Voronoi diagram. Several practical applications, however, solely exploit combinatorial properties of the Delaunay triangulation. This has led to the investigation of methods for constructing this structure directly, thus avoiding the need to calculate and store the coordinates of Voronoi vertices.

Lawson [1972] proposed an algorithm for constructing locally equiangular triangulations by local improvement. Starting with an arbitrary triangulation of the sites, edges are "flipped" according to the equiangularity criterion (stated in Section 1.2.4) until no more such exchanges are required. Sibson [1977] succeeded in proving that this procedure



ɜ 19. Equiangularity and empty circle prop-

ally constructs the Delaunay tri-
ılation. Lawson's original method
somewhat speeded up by Lee and
chter [1980] who proceed incremen-
and apply the edge flipping proce-
after each insertion of a new site.
ain [1976] uses the empty circle
erty to construct the Delaunay trian-
ion by successively adding triangles
e circumcircles are empty of sites.
ct, Figure 19 illustrates the equiva-
of the empty circle property and
equiangularity criterion. Although
simple to implement, all these al-
hms have an $O(n^2)$-time behavior in
orst case. On the other hand, Maus
] reports that a modification of
ain's algorithm based on prepartі-
ng exhibits an $O(n)$ expected run-
time for uniformly distributed sites.
and Schachter [1980] gave an
og n)-time Delaunay triangulation
ithm; it uses divide and conquer and
nilar to Shamos and Hoey's primal
noi algorithm. A modified imple-
ation using prepartitioning was
n by Dwyer [1987] to run in
og log n) expected time while re-
ng the optimal $O(n \log n)$ worst-case
it generalizes to the L_p-metrics for
2. Drysdale [1990] modifies the
e-and-conquer approach to compute,
$n \log n$) time, Delaunay triangula-
generated by general convex dis-
functions. A recent algorithm by
as et al. [1990] inserts (point) sites
ndom order. Using a structure simi-
the Delaunay tree [Boissonnat and
ud 1986], earlier versions of the

triangulation are maintained to facilitate the determination of the triangle covering the next site to be inserted. This practical algorithm exhibits a randomized running time of $O(n \log n)$ and is similar in spirit to Clarkson and Shor's [1988] convex hull algorithm.

The question arises as to which of the planar Delaunay algorithms can be extended to higher dimensions. We have already seen some early (although widely used) algorithms at the beginning of this section. It should be further observed that the method of higher dimensional embedding actually first constructs the Delaunay triangulation (via a convex hull), then derives the Voronoi diagram via dualization. Recently, Rajan [1991] proved a property of d-dimensional Delaunay triangulations related to equiangularity. It implies an incremental algorithm that makes the triangulation locally Delaunay after each insertion of a new site by applying a triangle-flipping procedure similar to Lawson's method in the plane. The algorithm is worst-case optimal for odd d. Whether every d-dimensional triangulation can be made Delaunay by local improvement remains open.

1.3.7 Storage Representation and Dynamization

The Voronoi diagram may be viewed as a data structure that organizes its defining sites in a prescribed manner. A host of applications of this data structure in computational geometry will be given in Section 2. Out of a number of possibilities to store the planar Voronoi diagram or, equivalently, its dual, the Delaunay triangulation, the *quad-edge structure* proposed by Guibas and Stolfi [1985] is particularly practical. Essentially, each edge is stored as a pair of directed edges. Each directed edge, in turn, stores the vertex it originates from and pointers to the previous and to the next (directed) edge of the region to its left. The quad-edge data structure provides a clean separation between topological and geometrical aspects and supports the implementation of standard techniques for

computing Voronoi diagrams, such as divide and conquer or incremental insertion. The structure has been generalized to higher dimensions by Dobkin and Laszlo [1989] (facet-edge structure in R^3) and by Brisson [1989] (cell-tuple structure in R^d). These data structures represent the incidence and ordering information in a cell complex in a simple uniform way. See also Aurenhammer and Edelsbrunner [1984] and Edelsbrunner et al. [1986] for different data structures representing cell complexes.

Like any data structure, the Voronoi diagram can be *dynamized*, that is, maintained for a set of sites that varies over time by insertion or deletion. The case of site insertion is covered by the algorithms mentioned before that build up the diagram on line by introducing new sites. Integration of a new region into the Voronoi diagram for n sites in the plane clearly costs $\Omega(n)$ time in the worst case since the region may have n edges. Gowda et al. [1983] handle insertions and deletions of sites in $O(n)$ time by means of the so-called *Voronoi-tree* that occupies $O(n \log\log n)$ space. This tree records the history of a divide-and-conquer construction of the diagram (Figure 20). Its leaves hold the sites in lexicographical order. Each inner node is associated with the diagram that comes from combining its sons' subdiagrams. The idea originates with Overmars [1981] who, however, did not give details of the method or its storage requirement. By a result by Aggarwal et al. [1989a], deletion of a given site can be performed in time proportional to the number of edges of its region.

Using the Delaunay triangulation, a planar set of sites can be organized into a hierarchical data structure, called the *Delaunay tree* by Boissonnat and Teillaud [1986]. This structure is semidynamic; it allows the insertion of sites and supports the retrieval of the triangle in which a given point falls in efficient expected and worst-case time. The Delaunay tree reflects the history of an incremental construction of the Delaunay triangulation (Figure 21). Any triangle T destroyed by inserting a new site gets as sons (solid pointers) the new triangles sharing an edge with T. Remaining triangles adjacent to T in such an edge get the respective new triangle as a stepson (dotted pointers). Guibas et al. [1990] further developed this structure. Among other results, they showed that the expected number of structural changes of the Delaunay tree is $O(n)$ if n sites are inserted in random order into an (initially empty) Delaunay triangulation.

A different way of dynamizing Voronoi diagrams or Delaunay triangulations is to allow *movement of sites* (continuous updates) rather than their insertion or deletion (discrete updates). The Voronoi diagram of continuously moving sites will change its shape continuously, although only at "critical" points in time will the combinatorial structure of the diagram change. Aside from degenerate cases, such combinatorial changes always are local (Figure 22). An edge between two regions (here 2 and 4) collapses to a vertex and reappears between two others (1 and 3) just in the moment when the four sites involved get cocircular. In the dual environment, this process corresponds to flipping the diagonals of the quadrilateral spanned by these four sites, which is the union of two Delaunay triangles. This suggests the definition of the *Delaunay history* of a set of moving sites as the chronologically ordered list of edge flips in their Delaunay triangulation. Tokuyama [1988] considered the case of two rigidly moving sets of n and m sites, respectively. Their Delaunay history has length $O(nm)$ and can be computed and preprocessed in $O(nm \log nm)$ time in order to retrieve the Delaunay triangulation (and the Voronoi diagram) in $O(n + m)$ time at any given moment. For n sites moving in fixed but individual directions and with constant individual velocity, the Delaunay history has length $O(n^3)$. This result by Imai et al. [1989] should be contrasted with the intuitive argument that $\binom{n}{4}$ cocircularities may

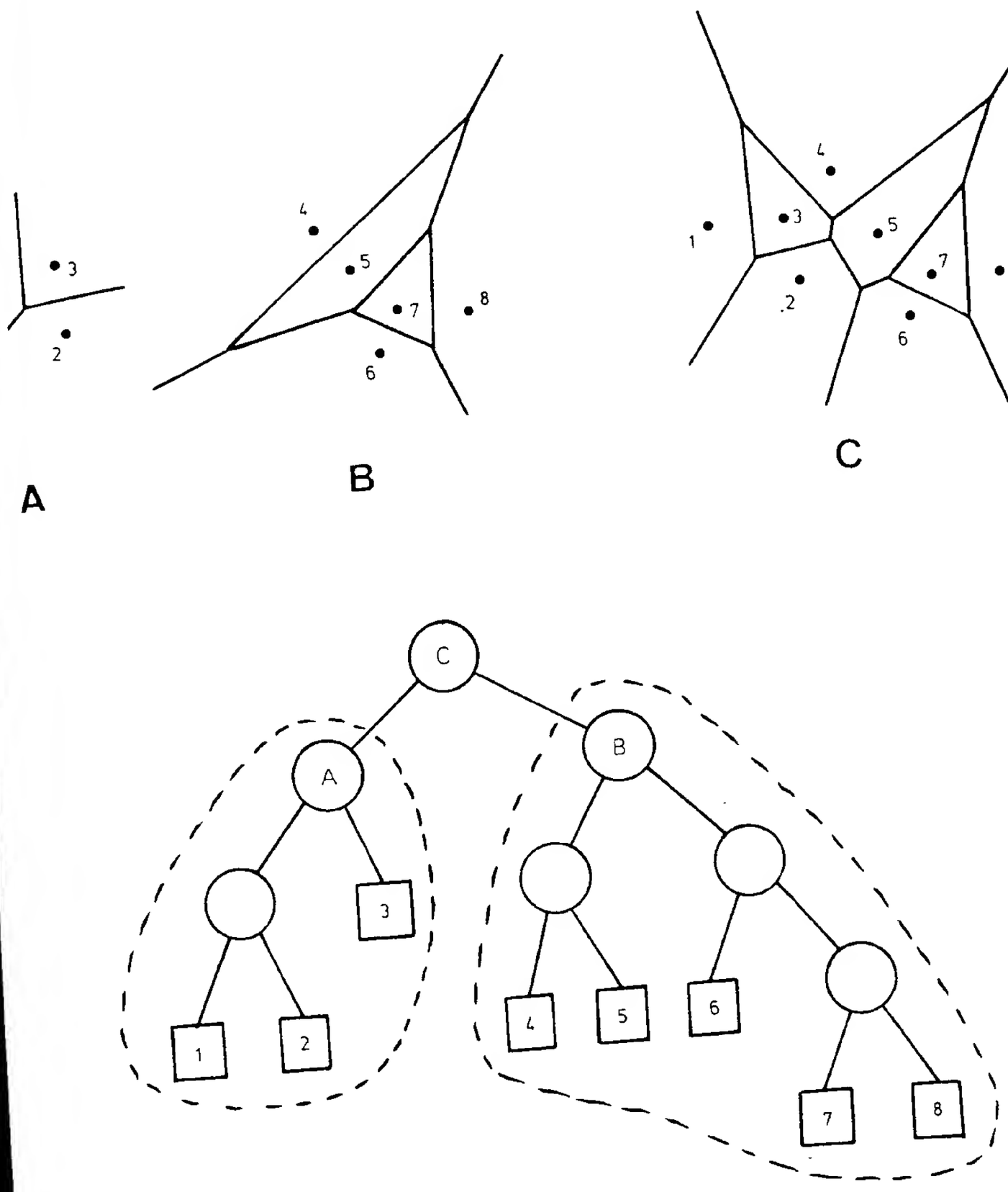
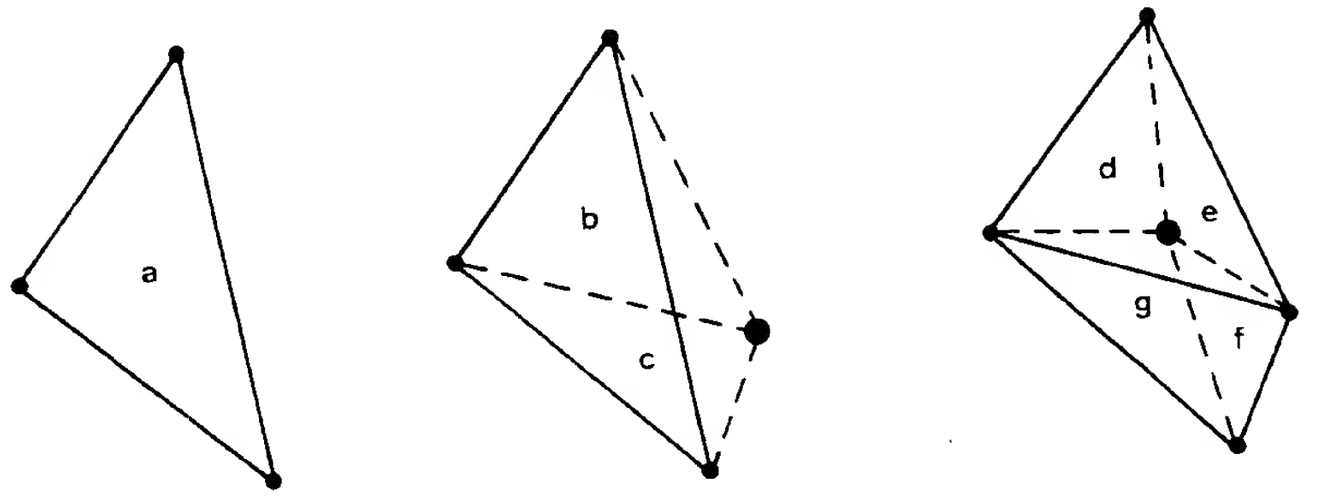


Figure 20. Voronoi tree [Gowda et al. 1983].

r among n sites. Guibas et al. [1991] ved that the bound remains nearly : if the sites are moving continu- / along rather general trajectories. also Aonuma et al. [1990] for related lts.

: Computing Voronoi Diagrams in Parallel

llelizing algorithms in computa- il geometry usually is a complicated task since many of the techniques used (incremental insertion or plane sweep, for instance) seem inherently sequential. Since it is one of the fundamental structures in this area, the Voronoi diagram has been among the first geometrical structures whose construction has been parallelized. The underlying model of computation mostly has been the CREW PRAM (concurrent-read, exclusive-write parallel random access machine), where



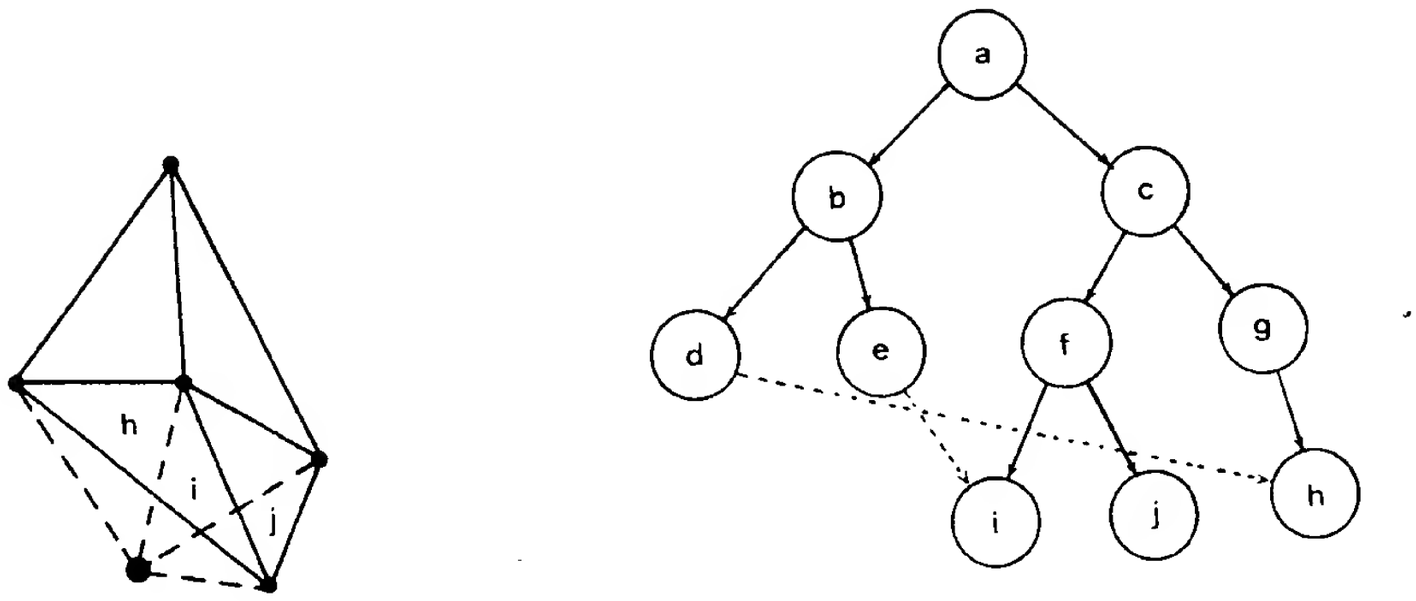


Figure 21. Delaunay tree.

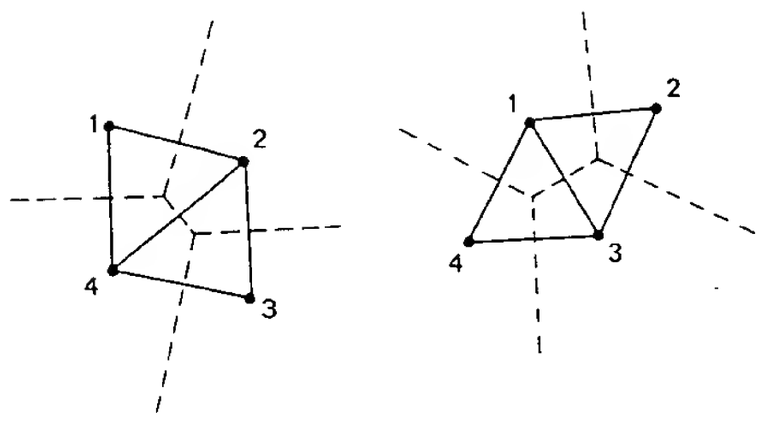


Figure 22. Combinatorial change.

processors share a common memory that allows concurrent reads but no simultaneous writes to the same cell.

As has been mentioned, $\Omega(n \log n)$ is a lower time bound for sequentially constructing the Voronoi diagram for n sites in the plane. Therefore, $O(\log n)$ parallel time is the best that can be achieved when only $O(n)$ processors are to be used. In her pioneering work, Chow [1980] showed that convex hulls in three space can be computed in $O(\log^3 n)$ time using $O(n)$ processors. By their duality to planar Voronoi diagrams, the complexity carries over. Aggarwal et al. [1988] improved the Voronoi diagram construction to $O(\log^2 n)$ parallel time by applying divide and conquer. Their approach critically depends on restricting the domain that can contain the "merge chain" (compare Figure 17) and has been improved recently by Cole et al. [1990] using a tailor-made data structure. An optimal parallel construction of Voronoi diagrams is still outstanding, however.

Goodrich et al. [1989] construct the Voronoi diagram for line segment sites in $O(\log^2)$ time using $O(n)$ processors. Their approach to constructing this important generalization is similar in spirit to Yap's [1987] curve segment algorithm. As a related question, Schwarzkopf

[1989] addresses the problem of computing the digitized image of a planar Voronoi diagram using certain processor networks.

2. ALGORITHMIC APPLICATIONS

The precomputation of a Voronoi diagram is the initial step of various algorithms in computational geometry. We have already mentioned some of them in the Introduction. In this part of the survey we demonstrate the broad scope of algorithmic applications of Voronoi diagrams and of closely related structures.

2.1 Closest-Site Problems

According to their definition, Voronoi diagrams apply naturally to various proximity problems. Determining closest sites plays a dominant role in this context since it appears as a subroutine of many geometric algorithms. Applications include clustering and contouring and various other problems whose relation to proximity may not be so obvious.

2.1.1 Nearest-Neighbor Queries

Probably the most popular problem in this area is the *post-office problem*. How should a fixed set of n sites in the plane be preprocessed in order to determine quickly the site closest to an arbitrarily chosen point (the query point)? The post-office problem is motivated in the Introduction as a two-dimensional file-searching problem. Another important application stems from a common data classification rule that requires assignment of new data points to the same class in which their nearest neighbor lies. An early solution yielding $O(\log n)$ query time but requiring $O(n^2)$ storage was provided by Dobkin and Lipton [1976]. Shamos [1975] pointed out that the Voronoi diagram of the sites divides the plane into regions of equal answer with respect to the post-office problem: By definition, the region of a site contains all points closer to this site than to all other sites. This reduces the problem to finding the region that contains the query point, an approach that became powerful once efficient data structures were developed for the latter problem: point location in planar straight-line graphs [Kirkpatrick 1983; Edahiro et al. 1984; and Edelsbrunner et al. 1986]. Actually, logarithmic query time and linear storage are achieved. This is asymptotically optimal since it matches the information-theoretical lower bound.

Shamos' approach in conjunction with Kirkpatrick's was worked out in a geometrically dual setting by Edelsbrunner and Maurer [1985]. Usage of a structure related to Boissonnat and Teillaud's [1986] Delaunay tree even obviates the need of postprocessing the Voronoi diagram for point location; Guibas et al. [1990] show that post-office queries are supported in $O(\log^2 n)$ expected time by this structure. An off-line solution to the post-office problem (all query points are given in advance rather than being specified on-line) was presented by Lee and Yang [1979].

In principle, higher dimensional variants of the post-office problem may be solved via the Voronoi diagram, too. Their practical relevance is evident from the equivalent file-searching problem for multiattribute data mentioned in the Introduction. Although Chazelle [1985] showed that point-location in a Voronoi diagram in R^3 can be performed efficiently, the approach suffers from a considerable storage overhead of $\Theta(n^2)$ that comes from the worst-case size of the diagram. In $d > 3$ dimensions, the prohibitively large size of the diagram may be reduced to some extent for that purpose [Dewdney 1977].

Generalizations of the post-office problem for nonpoint sites or for modified distance functions may be solved by means of an appropriate Voronoi diagram as well. We particularly mention Mount [1986] (geodesic distance on a polyhedral surface) and Aurenhammer et al. [1991] (probabilistic distances in the plane) as two generalizations leading to interesting types of diagrams.

2.1.2 *k*-Nearest-Neighbor Queries

An obvious and important generalization of the post-office problem is to ask for the k *closest sites* for a query point. It was observed by Shamos and Hoey [1975] that the order-k Voronoi diagram divides the plane into regions of equal answer with respect to this problem. Postprocessing the diagram for point-location yields a data structure that handles queries in $O(\log n + k)$ time and requires $O(k^2 n)$ storage. [The diagram may contain $\Theta(k(n-k))$ regions for each of which the k closest sites are stored.] If k is specified in the input, an $O(n^3)$-space data structure yielding the same query time was obtained by Edelsbrunner et al. [1986] via embedding the whole family of order-k Voronoi diagrams for the sites into an arrangement of planes. A technique for compacting order-k Voronoi diagrams recently developed by Aggarwal et al. [1990] drastically reduces the storage requirement of these approaches [to $O(n)$ for fixed k and to $O(n \log n)$ if k is in the input] while still achieving the optimal $O(\log n + k)$ query time.

A related problem is the *circular retrieval* problem that requires a report of all sites covered by a given query disk. In facility location, for example, one might be interested in all sites being influenced by a new facility of given radius of attraction. First steps toward a satisfactory solution were undertaken by Bentley and Maurer [1979]. Their method is based on a certain sequence of order-k Voronoi diagrams of the given sites and has been refined by Chazelle et al. [1986b] and recently by Aggarwal et al. [1990]. The latter refinement yields a data structure using $O(n \log n)$ space and achieving $O(\log n + t)$ query time, where t is the number of sites in the query disk.

2.1.3 Closest Pairs

Finding the *closest pair* among n sites certainly belongs to the most fundamental proximity problems. One obvious application arises in collision detection; the two closest sites are in the greatest danger of collision. In other applications, for example, in clustering, one needs to find the closest site for each of the sites. The straightforward approach clearly results in calculating $\Theta(n^2)$ distances. In the planar case, both questions were settled in optimal time $O(n \log n)$ by Shamos and Hoey [1975] by precomputing the Voronoi diagram of the sites. See Figure 23 depicting all closest pairs by bold segments. Since the region of a site and the region of its closest site always share an edge, one only needs to visit each edge and calculate the distance between the two sites whose separator yields that edge. This clearly takes $O(n)$ time once the diagram is available. Recently, Hinrichs et al. [1988a, 1988b] showed that it suffices to maintain certain parts of the diagram during a plane sweep rather than to compute the diagram explicitly in order to solve both problems in optimal time. Curiously, the *furthest pair* of sites cannot be derived directly from the furthest site Voronoi diagram but requires some additional point-location [Toussaint and Bhattacharya 1981]. Finding the furthest pair means determining the diameter of the underlying set of sites and thus has several applications.

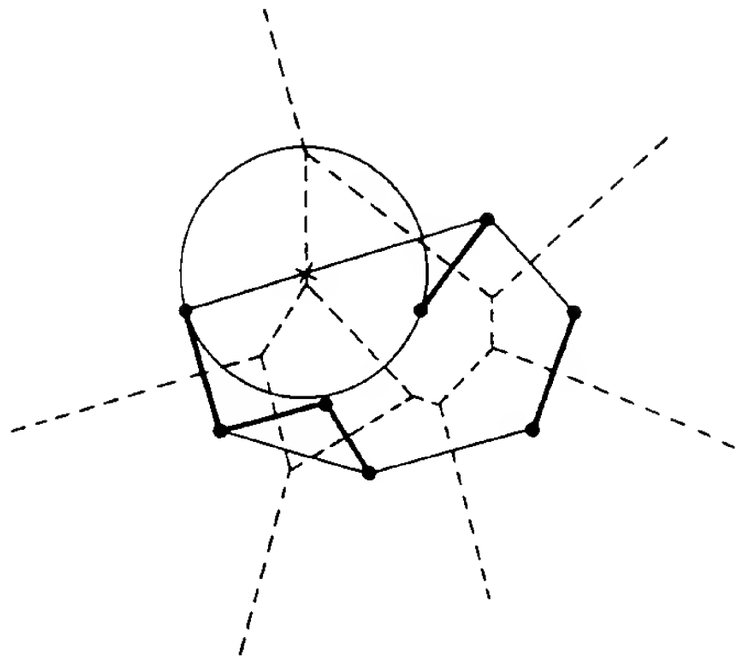


Figure 23. All closest pairs and the largest empty circle.

2.2 Placement and Motion Planning

Voronoi diagrams have the property that each point on a Voronoi edge maximizes the distance to the closest sites. Although being straightforward, this property is essential for the usefulness of Voronoi diagrams in placement and motion planning problems.

2.2.1 Largest Empty Figures

The *largest empty figure* problem is the following: Given n point sites in the plane, where should a figure of prescribed shape be placed so that its area is maximized but no site is covered? If the figure is a circle, the problem can be reformulated as finding a new site being as far as possible from n existing ones. Shamos and Hoey [1975] mentioned the relevance of this facility location problem to operations research and industrial engineering. They also showed how to find a largest empty circle in $O(n \log n)$ time: The center of such a circle is either a vertex of the Voronoi diagram of the sites or the intersection of a Voronoi edge and the boundary of the convex hull of the sites; see Figure 23 illustrating the occurrence of the latter case. Thus attention can be restricted to $O(n)$ possible placements of the circle.

The *smallest enclosing* circle of the sites is obtainable in $O(n \log n)$ time in a similar manner. Shamos [1978] claimed that this circle always will be centered at a vertex of the furthest site Voronoi diagram. This was proved true by Toussaint and Bhattacharya [1981]. From Figure 24 it can be seen that the edges of this diagram form a tree structure. This restricts the maximum number of vertices, and thus of placements of the circle, to $n - 2$.[7] On both problems, there exists

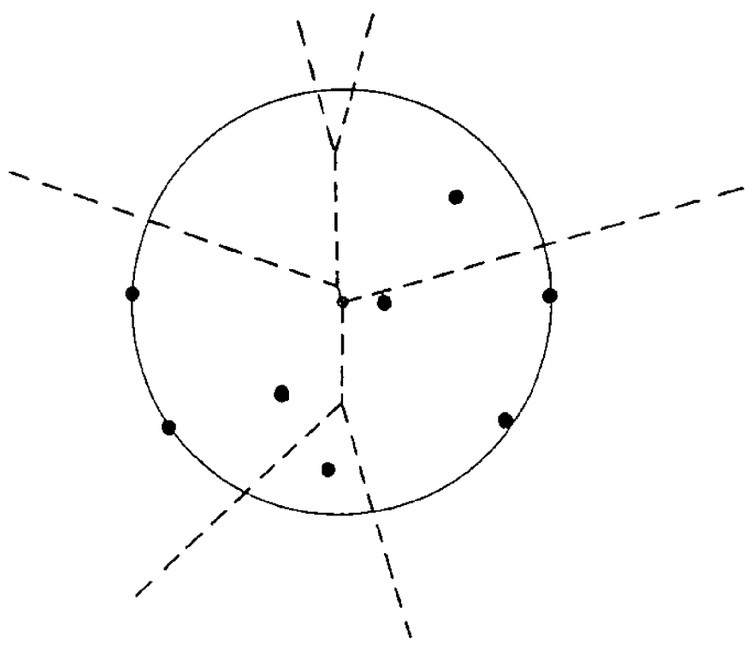


Figure 24. Furthest site Voronoi diagram and smallest enclosing circle.

a considerable literature [Preparata and Shamos 1985]. In particular, the problems are solvable in $O(n)$ time by means of linear programming as was demonstrated by Megiddo [1983].

The Voronoi diagram approach generalizes—while still achieving the $O(n \log n)$-time bound—to other figures provided these are "circles" with respect to the underlying distance function. For example, the largest empty axis-parallel square can be found using the Voronoi diagram in the L_∞-metric. Chew and Drysdale [1985] find largest empty *homothetic copies* of generally shaped convex figures via diagrams that are induced by a rather general class of distance functions. The determination of the largest empty axis-parallel rectangle turns out to be somewhat more complicated; the aspect ratio of the rectangle, and thus the distance function to be used, is not known in advance. This problem has an obvious application. If we are given a piece of fabric or sheet metal and the sites are flaws, what is the largest area rectangular piece that can be salvaged? Chazelle et al. [1986a] gave an $O(n \log^3 n)$-time solution. They use an interesting type of Voronoi diagram that can be intepreted as the power diagram of a certain set of circles [Edelsbrunner and Seidel 1986; Aurenhammer and Imai 1988]. The running time above was

[7] It is easily seen that the regions of the furthest site diagram are either unbounded or empty. The corresponding *furthest site Delaunay triangulation* thus is an outer planar graph (with at most n sites as vertices), which is well known to contain at most $2n - 3$ edges and $n - 2$ triangles.

improved to $O(n \log n)$ by Aggarwal and Suri [1987] using different methods. The properties of the Voronoi diagram for line segment sites have been exploited by Chew and Kedem[1989] to find the largest *similar copy* of a convex polygon such that no segment is overlapped. As a related question, Aonuma et al. [1990] place a given convex polygon inside and as far as possible from the boundary of an arbitrary polygon. Their algorithm relies on Voronoi diagrams dynamized with respect to certain site movements.

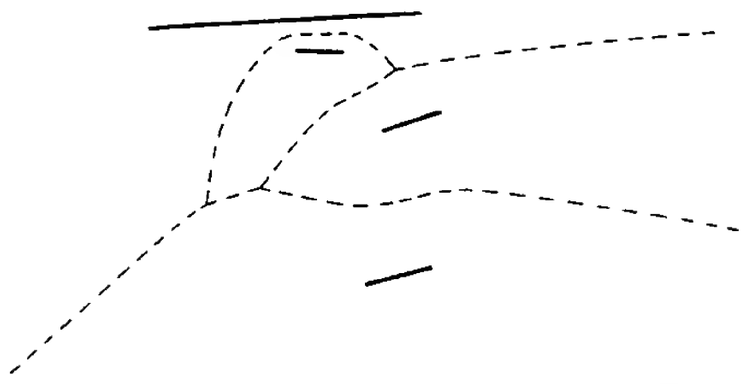


Figure 25. Voronoi diagram for line segments.

2.2.2 Translational Motions

Voronoi diagrams are an aid in the planning of *collision-free motions* of a figure in the presence of obstacle sites. Problems of this kind arise in robotics. The Voronoi diagram approach plays an important role in this area and is called the retraction method. Intuitively speaking, when moving on the edges of the Voronoi diagram the robot always keeps as far as possible from the neighboring obstacles. This was first perceived by Rowat [1979]; several authors pursued this idea further.

For many applications it is feasible to approximate the scenario of obstacles by polygonal sets. O'Dunlaing and Yap [1985] plan the motion of a disk between two given points in this environment. In a preprocessing step, the Voronoi diagram of the obstacles is computed by taking their n boundary segments as its defining sites. The graph formed by the $O(n)$ straight or parabolic edges of the diagram can be computed in $O(n \log n)$ time [Kirkpatrick 1979; Fortune 1987]. Figure 25 depicts the graph (dashed) for four line segment obstacles (bold). Observe that each point on an edge is equidistant from its two closest obstacles. Therefore, for each edge, it is easy to compute the minimum clearance when moving on it. To produce a collision-free motion (if it exists), one first finds the edges e and e' closest to the center of the disk in initial and final position, respectively, and then searches the graph starting with e until either e' is reached or the clearance falls short of the disk radius. All of that can be done in additional $O(n)$ time.

Compared to others, this simple approach has the additional advantage that the disk radius need not be known during the preprocessing phase. Extending the approach, Leven and Sharir [1987] plan translational motions of a convex planar robot in $O(n \log n)$ time. Note that generally shaped robots may be circumscribed by a convex figure, and paths feasible for the figure will do for the robot as well.

2.2.3 Rotational Motion

Allowing the robot to rotate adds a third degree of freedom to the problem. Much attention has been paid to moving—amidst polygonal obstacles—a line segment that is allowed to rotate. The position of a segment (of given length) in the plane is determined by the triple $p = (x, y, \varphi)$ indicating the coordinates of one endpoint and the orientation. The set of all triples p such that the corresponding segment $L(p)$ does not collide with any obstacle is a subset of the underlying three-dimensional configuration space. Within this subset, we may define a Voronoi diagram as the set of all p such that $L(p)$ is equidistant from its two closest obstacles; closeness is with respect to the minimum distance between segment and obstacle.

O'Dunlaing et al. [1986, 1987] provided a thorough analysis of this diagram (which seems to belong to one of the most complex types). In particular, they showed that the diagram can be

constructed, and a motion be planned in $O(n^2 \log n \log^* n)$ time, thus being a significant improvement over earlier techniques. Sifrony and Sharir [1986] speeded up the method—by introducing a related graph—to run in $O(t \log n)$ time, where $t = O(n^2)$ is a parameter dependent on the segment length. Canny and Donald [1988] simplified the method by relaxing the definition of the diagram.

There are several other versions of the motion planning problem to which Voronoi diagrams apply [Rohnert 1988]. An axiomatic characterization of the boundary properties of a generally shaped three-dimensional scenaro in order to define a Voronoi diagram suited to motion planning was proposed by Stifter [1989]. For a more detailed introduction and review of the use of Voronoi diagrams in motion planning, the interested reader may consult Schwartz and Yap [1986] or Alt and Yap [1990].

2.2.4 *Path Planning*

A special case of planning collision-free motions, where the robot is taken to be a single moving point, is *path planning*. Finding euclidean shortest paths in the presence of polygonal (or polyhedral) obstacles has received particular interest. For example, consider the problem of determining that location, for each of m fixed locations within a polygonal factory floor, reachable on the shortest path. This problem can be solved in $O(n + m)\log^2(n + m)$ time by first constructing the Voronoi diagram of n sites under the geodesic metric interior to a simple polygon with m edges [Aronov 1989]. The site yielding the shortest path to a particular site defines a neighboring region in the diagram. This improves an earlier result by Asano and Asano [1986]. Related questions can be solved by means of the furthest site geodesic Voronoi diagram inside a polygon; Aronov et al. [1988] showed how to construct this structure efficiently.

The following important question is still unsolved. Is it possible to compute the geodesic Voronoi diagram for n sites

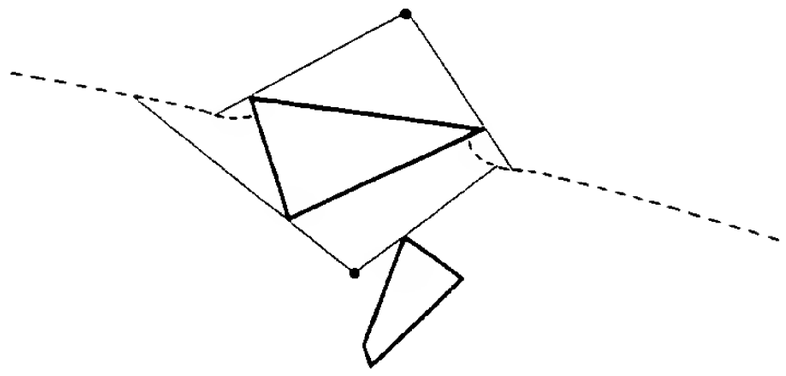


Figure 26. Geodesic separator.

and a collection of disjoint and convex polygons with a total of m edges in $O(n + m)\log(n + m)$ time? The main problem in this respect is that computing geodesic distances is not a constant-time operation. Figure 26 shows the separator (dashed) of two sites under the geodesic distance among two convex polygons (bold). It is composed of straight line and hyperbola segments meeting at intersections with visibility lines (solid).

Path-planning problems in three-dimensional polyhedral spaces have been studied, by among other authors, Mount [1986], Mitchell et al. [1987], and Baltsan and Sharir [1988]. For example, finding shortest paths lying entirely on a (generally nonconvex) polyhedral surface is of interest in areas like autonomous vehicle navigation, where hilly terrains are being modeled. Efficient solutions are obtained by using Voronoi diagrams defined in such spaces. Moreover, since finding shortest paths is closely related to determining visibility between obstacles, Voronoi diagrams involving visibility constraints have been exploited in this context. An example is the peeper's Voronoi diagram used by Baltsan and Sharir [1988] and by Aronov [1989]. Each point in the plane is assigned to the closest visible site, and visibility is constrained to a segment on a line avoiding the convex hull of the sites (see Figure 36). This structure can attain a size of $\Theta(n^2)$ and is constructable in $O(n^2)$ time [Aurenhammer and Stöckl 1988].

2.3 Triangulating Sites

Intuitively speaking, *triangulating* n sites in the plane means partitioning the

plane into triangles whose vertices are all and only the sites. Triangulations are important because they (as being planar graphs) on the one hand have only linear size and, on the other hand, establish connectivity information among the sites that suffices for many applications.

2.3.1 Equiangular Triangulations

An early and major application is the interpolation of functions of two variables, where function values initially are known only at irregularly placed sites. Practical evidence stems from finite element methods or from processing graphical data, for example, in terrain modeling. Given a triangulation of the sites, the function value at an arbitrary point can be computed by interpolation within the triangle containing that point. Lawson [1972] and McLain [1976] have reported that a triangulation is well suited to interpolation if its triangles are nearly equiangular. The Delaunay triangulation is the unique triangulation that is optimum in this sense [Sibson 1977]. Sibson [1980] further suggests the use of local coordinates obtained from sites being adjacent in this triangulation in order to compute smooth surfaces. There might, however, occur more-sided polygons (rather than triangles) if various sites are cocircular. Mount and Saalfeld [1988] showed how to triangulate such polygons in order to retain equiangularity. This process does not affect the $O(n \log n)$ time needed for constructing the Delaunay triangulation.

Given a set of sites and a particular function value (height) at each site, any triangulation of the sites defines a triangular surface in space. The "roughness" of such a surface T may be measured by the *Sobolev seminorm*

$$\sum_{\Delta \in T} |\Delta| \left(\alpha_\Delta^2 + \beta_\Delta^2\right),$$

where $|\Delta|$ denotes the area of the spatial triangle Δ, and α_Δ and β_Δ denote the slopes of the plane containing Δ. By exploiting equiangularity, Rippa [1990] showed that the Sobolev seminorm is minimized if the underlying triangulation of the sites is Delaunay. This result is somewhat surprising since the Delaunay triangulation—although itself clearly being independent of the height at each site—optimizes a quantity that depends on these heights.

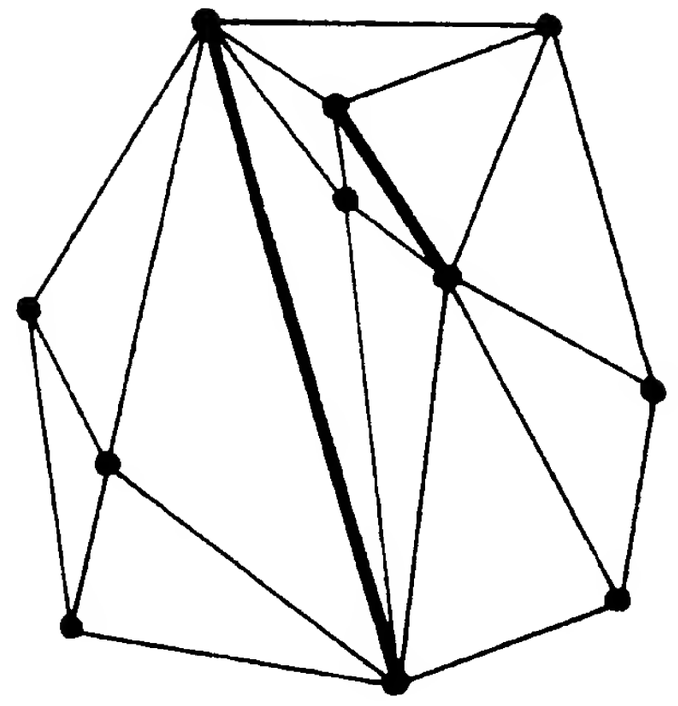


Figure 27. Constrained Delaunay triangulation.

2.3.2 Constrained Triangulations

In a *constrained* Delaunay triangulation, prescribed edges are forced in as part of the triangulation. See Section 3.3 for a formal definition. Figure 27 depicts such a triangulation with two prescribed edges (bold). This structure can be constructed in $O(n \log n)$ time [Seidel 1988] and has two major applications. First, it provides a more realistic approach to modeling terrain surfaces since salient elements like mountain ridges or valleys may be prescribed. Lee and Lin [1986] showed that the equiangularity criterion is still fulfilled by constrained Delaunay triangulations. Since triangles with small angles produce a poor computer graphics display, the best possible visualization is achieved. Triangulations inside a simple polygon are an important special case. The boundary edges of the polygon will not be edges of the (unconstrained) Delaunay triangulation of the polygon vertices, in general, and thus have

to be prescribed.[8] Chew [1989b] used constrained Delaunay triangulations to generate triangular meshes inside polygons where all angles are between 30° and 120°.

Second, constrained Delaunay triangulations are an aid in path planning. In order to plan a path from A to B that avoids a set of n line segment obstacles (and thus arbitrarily shaped polygonal regions, for instance) one might construct this triangulation—by taking A and B and the segment endpoints as sites and the segments as prescribed edges—and then search the graph formed by triangulation edges. This was observed by Chew [1986], who also proved the following interesting result: Between any two sites there is a path in this triangulation whose length is at most $\sqrt{10}$ times the geodesic distance between them, provided the L_1-metric (instead of the euclidean) is taken to define the triangulation. This yields an $O(n \log n)$-time algorithm for approximating the optimal path, which is a significant improvement over exact methods.

Dobkin et al. [1990] proved a factor of about 5 for (nonconstrained) euclidean Delaunay triangulations and the euclidean distance. Their result has been improved to a factor of about $\frac{5}{2}$ by Keil and Gutwin [1989]. Results of this kind are rather surprising since they show the existence of a sparse graph being almost as "good" as the complete graph on n sites, independently of n.

Let us mention another application. Given n sites and some prescribed (noncrossing) edges between them, a diagonal is a new edge that does not cross any prescribed one. A popular heuristic for triangulations with minimum total edge length is the greedy method. This method iteratively adds shortest diagonals by considering the current (and initially empty) set of edges as prescribed. Lingas [1989] reports that a shortest diagonal can be found in $O(n)$ time given the constrained Delaunay triangulation. This gives an $O(n^2 \log n)$-time and $O(n)$-space algorithm for constructing such *greedy triangulations*. Previous methods required $O(n^2)$ space. Whether a minimum length triangulation for n sites in the plane can be computed in polynomial time is an important open question.

2.3.3 3D Triangulations

The problem of "triangulating" point sites in three-space arises in the decomposition of three-dimensional objects, for example, of computer-generated solid models or of objects specified by planar cross sections. Field [1986], Boissonnat [1988], and others report on advantages of tetrahedral decompositions and propose the use of the Delaunay triangulation. Although there is some evidence for the quality of thus produced tetrahedra, some important properties of the two-dimensional Delaunay triangulation do not carry over.

First, the number of tetrahedra may be $\Theta(n^2)$, hence leading to a storage requirement that is prohibitive for many applications. This fact suggests the use of postprocessing. Chazelle et al. [1990] showed the following somewhat counterintuitive result: The number of tetrahedra may always be lowered to about $n\sqrt{n}$ by adding about $\sqrt{n}$ new sites. This process will also eliminate elongated (and thus undesirable) tetrahedra. In fact, adding $O(n)$ (well-chosen) sites will always reduce the size of the triangulation to $O(n)$, with the additional effect that only constantly many edges emanate from each site [Bern et al. 1990].

Second, the question of finding a three-dimensional analog to the equiangularity property is still unsettled. Rajan [1991], however, recently succeeded in proving a related property. The three-dimensional Delaunay triangulation minimizes the largest *containment*

[8] It is an interesting question how many new sites have to be placed on the polygon boundary in order to force it in as a subgraph of the Delaunay triangulation. Schwarzkopf (personal communication, 1990) observed that $\Omega(n^2)$ sites may be necessary for an n-vertex polygon. A known upper bound is $2^{O(n \log n)}$. The $O(n)$ bound claimed in Boissonnat [1988] involves a constant depending on the angles of the polygon.

radius of the tetrahedra; that is, the radius of the smallest sphere containing a tetrahedron. This sphere coincides with the circumsphere if and only if its center lies within the tetrahedron. Note that the containment radius measures the tetrahedron size more appropriately than the circumradius since small but elongated tetrahedra may have arbitrarily large circumradii. The Delaunay triangulation is the most compact triangulation in this sense. This property is quite remarkable as—unlike the two-dimensional situation—the number of tetrahedra is not independent of the way of triangulating the given sites but may vary from linear to quadratic.

There is another property of Delaunay triangulations that turns out to be useful in three-dimensions. For a set of triangles in three-space, an in-front/behind relation may be defined with respect to a fixed viewpoint. Generalizing a result by De Floriani et al. [1988], Edelsbrunner [1989] proved that this relation is acyclic for the triangles in a three-dimensional Delaunay triangulation, no matter where the viewpoint is chosen. This is relevant to a popular algorithm in computer graphics that eliminates hidden surfaces by first ordering the three-dimensional objects with respect to the in-front/behind relation and then drawing them from back to front, thus over-painting invisible parts. For Delaunay triangles, such an ordering always will exist, which is quite important in view of their frequent use in practice. In particular, so-called α-shapes are made of such triangles. α-shapes have been used by Edelsbrunner et al. [1983] to model the shape of point sets.

2.4 Connectivity Graphs for Sites

The Delaunay triangulation contains, as subgraphs, various structures with far-ranging applications. We will briefly discuss some of them in the sequel.

2.4.1 Spanning Trees

Given n point sites in the plane, a (euclidean) *minimum spanning tree* is a straight-line connection of the sites with minimum total edge length. See Figure 8 for an illustration. Not surprisingly, this structure plays an important role in transportation problems, pattern recognition, and cluster analysis. Construction methods working on general weighted graphs have been known for a long time; see Kruskal [1956], Prim [1957], and, more recently, Yao [1975] whose algorithm works in $O(m \log\log n)$ time for a graph with m edges. All these algorithms use a basic fact: For any partition of the sites into two subsets, the shortest edge (edge of minimum weight) between the subsets will be present in the tree. Applying Yao's algorithm directly to the complete euclidean distance graph for the sites results in an $\Omega(n^2)$ running time, however.

Shamos and Hoey [1975] recognized that the edges of a minimum spanning tree must be Delaunay edges. This interesting and important property holds since, by the fact mentioned above, each tree edge is the diameter of a disk empty of sites and, by definition, each triangulation edge is a chord of some disk empty of sites. From the computational viewpoint, Yao's algorithm just needs to be applied to the Delaunay triangulation—a graph with $O(n)$ edges. In fact, $O(n)$ time suffices to derive the minimum spanning tree from the Delaunay triangulation [Preparata and Shamos 1985]. This gives an $O(n \log n)$-time algorithm, which is optimal by reduction to sorting n real numbers.[9]

The relationship above extends to more general metrics. For example, Hwang [1979] extends it to the L_1-metric (compare Figure 3). One may be tempted to conjecture that the (euclidean) *maximum spanning tree* can be obtained from the furthest site Delaunay triangulation of the sites (the dual of the furthest site

[9] The minimum spanning tree should not be confused with the *minimum Steiner tree* of a set of sites, a concept where the addition of new points is allowed in order to minimize the total edge length. The problem of constructing a minimum Steiner tree is NP-complete [Garey et al. 1976].

Voronoi diagram). Shamos [1978] showed that this conjecture is erroneous. Eppstein [1990], however, pointed out that two sites are connected in one of the k *smallest* spanning trees only if both belong to the same subset of sites defining a region of the order-$(k+1)$ Voronoi diagram. This leads to an improved algorithm for finding the k smallest spanning trees.

The efficient construction of the minimum spanning tree in three space has turned out to be a difficult problem. Precomputation of the Delaunay triangulation does not help in general since the complete graph may be the output in the worst case. Efficient approximation and expected-time algorithms were given, among other authors, by Vaidya [1988a] and Clarkson [1989]. The first algorithm running in subquadratic worst-case time is due to Yao [1982]; it can be speeded up to run in $O(n \log n)^{1.5}$ time by using Voronoi diagrams of carefully chosen groups of sites. By exploiting—in addition—an interesting relation to a certain closest point problem, Agarwal et al. [1990] were able to bring down the complexity to $O(n^{4/3} \log^{4/3} n)$. It still remains open whether an algorithm matching the best currently known lower bound of $\Omega(n \log n)$ can be developed.

2.4.2 Spanning Cycles

A graph connecting n sites is called *Hamiltonian* if it contains some cycle passing through all the sites. The question of whether the Delaunay triangulation in the plane is Hamiltonian arose in pattern recognition problems where a reasonable simple curve through the given sites is desired. Although counter-examples were given by Kantabutra [1983] and Dillencourt [1987a, 1987b], Delaunay triangulations have been used with success for this problem. This gives evidence that this graph is Hamiltonian with high probability. Dillencourt [1987c] supports this thesis by proving a result concerning the connectivity of Delaunay triangulations.

A (euclidean) *traveling salesman tour* is a minimum-length cycle spanned by n sites. Constructing traveling salesman tours is a prominent problem of combinatorial optimization. Papadimitriou [1977] showed its NP-completeness. The problem becomes tractable for tours being close to optimal. It follows from the above discussion that the Delaunay triangulation will not contain a traveling salesman tour, in general.[10] See also Figure 28 where the triangulation and the tour are drawn solid and dashed, respectively. Rosenkrantz et al. [1977], however, observed that traversing the minimum spanning tree twice will produce a tour that is within a factor of 2 of the optimum: Since removing an edge from the traveling salesman tour leaves a spanning tree of the sites, this tour must be longer than the minimum spanning tree. Note that this approximation algorithm takes only $O(n \log n)$ time. By partitioning the minimum spanning tree into paths, then constructing a minimum-length matching of their endpoints, Christofides [1976] improved this to a factor of 1.5, with the expense of an $O(n^2 \sqrt{n} \log^4 n)$ construction time due

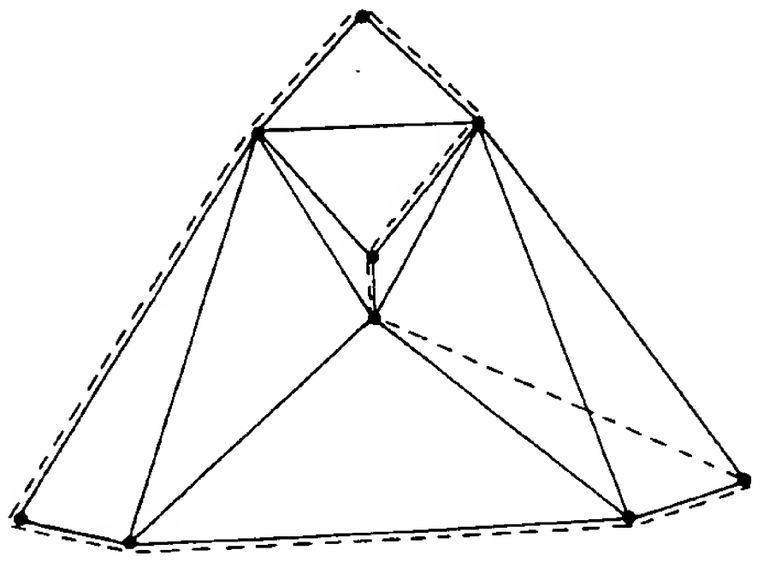


Figure 28. Delaunay triangulation and salesman tour [Dillencourt 1987].

[10] This is even true for special traveling salesman tours called *necklace tours* that are cycles realizable as the intersection graph of disks around the sites. Necklace tours can be found in polynomial time in case of their existence [Edelsbrunner et al. 1987].

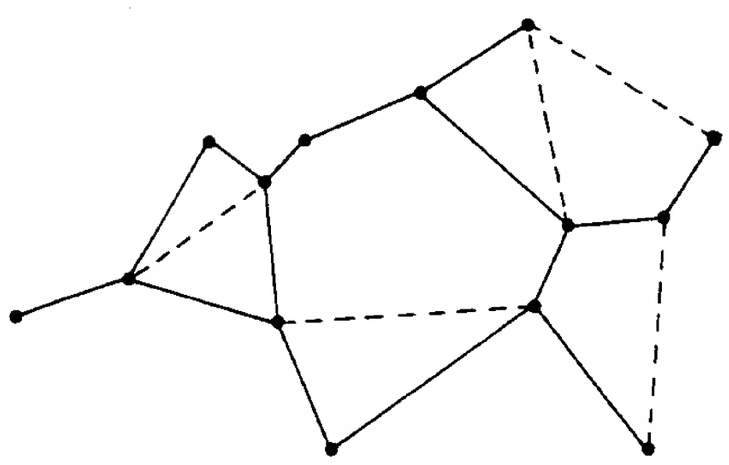


Figure 29. Relative neighborhood graph and Gabriel graph.

to the best known matching algorithm by Vaidya [1988b].[11]

2.4.3 Relative Neighborhood and Gabriel Graph

It is worth mentioning two graphs that are "in between" the minimum spanning tree and the Delaunay triangulation in the sense of making explicit more proximity information than does the former and less than does the latter. Among them is the *relative neighborhood graph* that connects two sites provided no other site is closer to both of them than their interpoint distance. In Figure 29, the edges of the relative neighborhood graph are shown solid. Toussaint [1980] reported the usefulness of this structure in pattern recognition, and Supowit [1983] succeeded in constructing it in $O(n \log n)$ time, given the Delaunay triangulation. Yao (personal communication, 1988) announced that $O(n)$ time suffices for deriving the relative neighborhood graph from the latter. Interestingly, this graph retains its linear size in higher dimensions (provided a general placement of the sites). This stresses the importance of its rapid construction, especially in three dimensions, since it would imply fast minimum spanning tree algorithms. Unfortunately, no algorithm with significantly subquadratic running time is known.

A similarly defined construct is the *Gabriel graph*. It contains an edge between two sites if the disk having that edge as its diameter is empty of sites. This concept proved useful in processing geographical data [Gabriel and Sokal 1969; Matula and Sokal 1980]. Howe [1978] observed that the Gabriel graph consists of just those Delaunay edges that intersect their dual Voronoi edges. Based on the same observation, Urquhart [1980] gave an $O(n \log n)$ time algorithm for its construction. It is easily seen that the Gabriel graph is a supergraph of the relative neighborhood graph. The edges of the former are shown solid or dashed in Figure 29.

2.5 Clustering Point Sites

Clustering a set of n point sites in the plane means determining partitions of the set that optimize some predefined clustering measure. This measure usually is a function of the interpoint distances of the set to be clustered.

2.5.1 Hierarchical Methods

A *single linkage clustering* [Hartigan 1975] hiearchically clusters the sites as follows. Initially, the sites are considered to be clusters themselves. As long as there is more than one cluster, the two closest clusters are merged. The distance of two clusters is defined as the minimum distance between any two sites, one from each cluster. This process can be carried out in a total of $O(n \log n)$ time by maintaining, at each stage, the following Voronoi diagram: Each point of the plane belongs to the region of the closest cluster, and closeness is with respect to the closest site within a cluster. The regions of the two closest clusters will always have a common edge in the diagram. That edge corresponds to an edge of the minimum spanning tree of the entire set of sites [Shamos and Hoey 1975].

[11] A *minimum-length matching* of $2n$ sites is a graph with n edges joining pairs of sites, such that the pairs are distinct and the total edge length is minimized. Akl [1983] showed that a minimum-length matching is not necessarily a subgraph of the Delaunay triangulations of the sites.

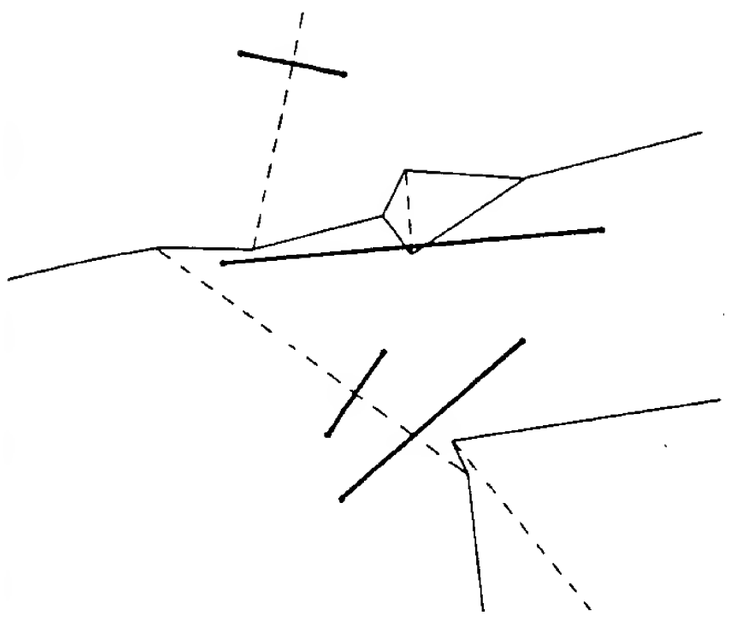


Figure 30. Complete linkage Voronoi diagram.

The model of *complete linkage clustering* is similar, except that the distance of two clusters is defined as the maximum distance between any pair of sites, one from each cluster. The corresponding Voronoi diagram, where closeness of a point in the plane is with respect to the furthest site in a cluster, deserves special attention. Its size is $\Omega(n^2)$ for general clusters and $O(n)$ provided the convex hulls of the clusters are pairwise disjoint [Edelsbrunner et al. 1989]. Figure 30 shows the diagram induced by four two-site clusters, indicated by bold segments. For each cluster, the portion of the bisector of its two sites that lies within its region is drawn dashed. The diagram can be constructed in $O(n \log n)$ randomized time for convex-hull disjoint and constant-sized clusters and—in this setting—applies to the retrieval of expected-nearest sites [Aurenhammer et al. 1991]. Efficient maintenance of the diagram during the clustering process would lead to an improved clustering algorithm but remains open.

2.5.2 *Partitional Strategies*

As another example, consider the problem of partitioning a set of n sites into a fixed number of t clusters such that the minimum of the (single linkage) distances between clusters is maximized. The following elegant $O(n \log n)$-time solution was proposed by Asano et al. [1988]. Construct the minimum spanning tree of the sites and remove its $t - 1$ longest edges. The resulting t subtrees already give the desired clustering. Using similar methods, Asano et al. [1988] obtain an equally efficient algorithm for partitioning the sites into two clusters so that their maximum diameter is minimized. This problem becomes considerably harder if partitions into more than two clusters are sought.[12] Partitions into *separable clusterings* have been investigated by Dehne and Noltemeier [1985] and Heusinger and Noltemeier [1989], who exploited the fact that clusters of k sites being separable from the remaining sites by a straight line define unbounded regions in the associated order-k Voronoi diagram.

2.5.3 *Optimum Cluster Selection*

The *k-variance problem* asks for selecting, from a given set S of n sites, a cluster of k sites with minimum variance, the sum of squares of all intersite distances in the cluster. Aggarwal et al. [1989b], who cite applications in pattern recognition, observed the following interesting fact. If C is k-cluster of minimal variance, then the region of C in the order-k Voronoi diagram of S is nonempty. Thus it suffices to examine the variances of sets of k sites associated with Voronoi regions. This can be done in time proportional to the number, $O(k(n - k))$, of regions by exploiting the fact that sets of neighbored regions differ in exactly two sites. Thus, the most time-consuming step is the precomputation of a higher order Voronoi diagram. A similar approach leads to improvements if the cluster measure to be minimized is not variance but diameter. If C is a k-cluster with minimum diameter, then C is contained in some set of $3k - 3$ sites whose region in the order-$(3k - 3)$ Voronoi diagram of S is nonempty;

[12] Asano et al. [1988] claimed its NP-completeness. Rote (personal communication, 1990), announced a polynomial algorithm in the case of three clusters.

see Aggarwal et al. [1989b] where an $O(k^2 n\sqrt{k}\log n)$-time selection algorithm is obtained in this way.

Several other clustering algorithms that rely on Voronoi diagrams have been described in the literature; some of them are mentioned in the Introduction.

3. SELECTED TOPICS

A thorough understanding of the geometric, combinatorial, and topological properties of Voronoi diagrams is crucial for the design of efficient construction algorithms. This third part of the survey presents these properties in a unified manner and discusses their algorithmic implications. We start with a description of the various relationships of Voronoi diagrams to objects in one more dimension. We continue with some topological properties that are particularly useful for the divide-and-conquer construction of a large class of Voronoi diagrams in the plane. Finally, we investigate a continuous deformation of the planar Voronoi diagram being well suited to construction by the plane-sweep technique and extend it to generalized Voronoi diagrams.

3.1 Geometry of Voronoi Diagrams: Their Relationship to Higher Dimensional Objects

Voronoi diagrams are intimately related to several central structures in discrete geometry. This section exhibits these relationships in a comprehensive manner and discusses some of their geometric and algorithmic consequences. Although our main interest is in the classical type of Voronoi diagram, it is advantageous to base the discussion on the more general concept of power diagram since the geometric correspondences to be described extend to that type in a natural way. We shall refer to d dimensions in order to point out the general validity of the results.

3.1.1 *Convex Polyhedral Surfaces*

Consider a set S of n point sites in R^d such that each site $p \in S$ is assigned an individual real number $w(p)$ called its *weight*. The *power function* of a point $x \in R^d$ with respect to p is specified by

$$\mathrm{pow}(x, p) = (x - p)^T(x - p) - w(p).$$

For $w(p) > 0$, one may think of the weighted site p as a sphere in R^d with radius $\sqrt{w(p)}$ around p; for a point x outside this sphere, $\mathrm{pow}(x, p) > 0$, and $\sqrt{\mathrm{pow}(x, p)}$ expresses the distance of x to the touching point of a line tangent to the sphere and through x. In the plane, the locus of equal power with respect to two weighted sites p and q is commonly called the *power line* of the corresponding circles. If the circles are nondisjoint, their power line passes through their points of intersection; see Figure 31. In general dimensions, the locus of equal power is a hyperplane in R^d, called the *chordale* of p and q or $\mathrm{chor}(p, q)$, for short. $\mathrm{chor}(p, q)$ is orthogonal to the straight line connecting p and q but does not necessarily separate p from q. For $w(p) = w(q)$, $\mathrm{chor}(p, q)$ degenerates to the symmetry hyperplane of p and q. Let $h(p, q)$ denote the closed halfspace bounded by $\mathrm{chor}(p, q)$ and containing the

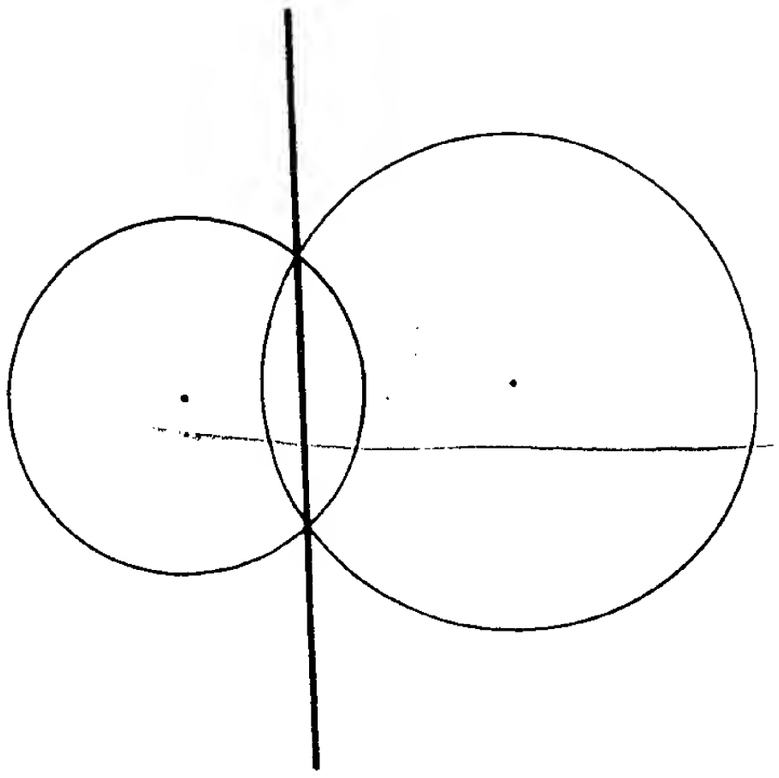


Figure 31. Power line of two circles.

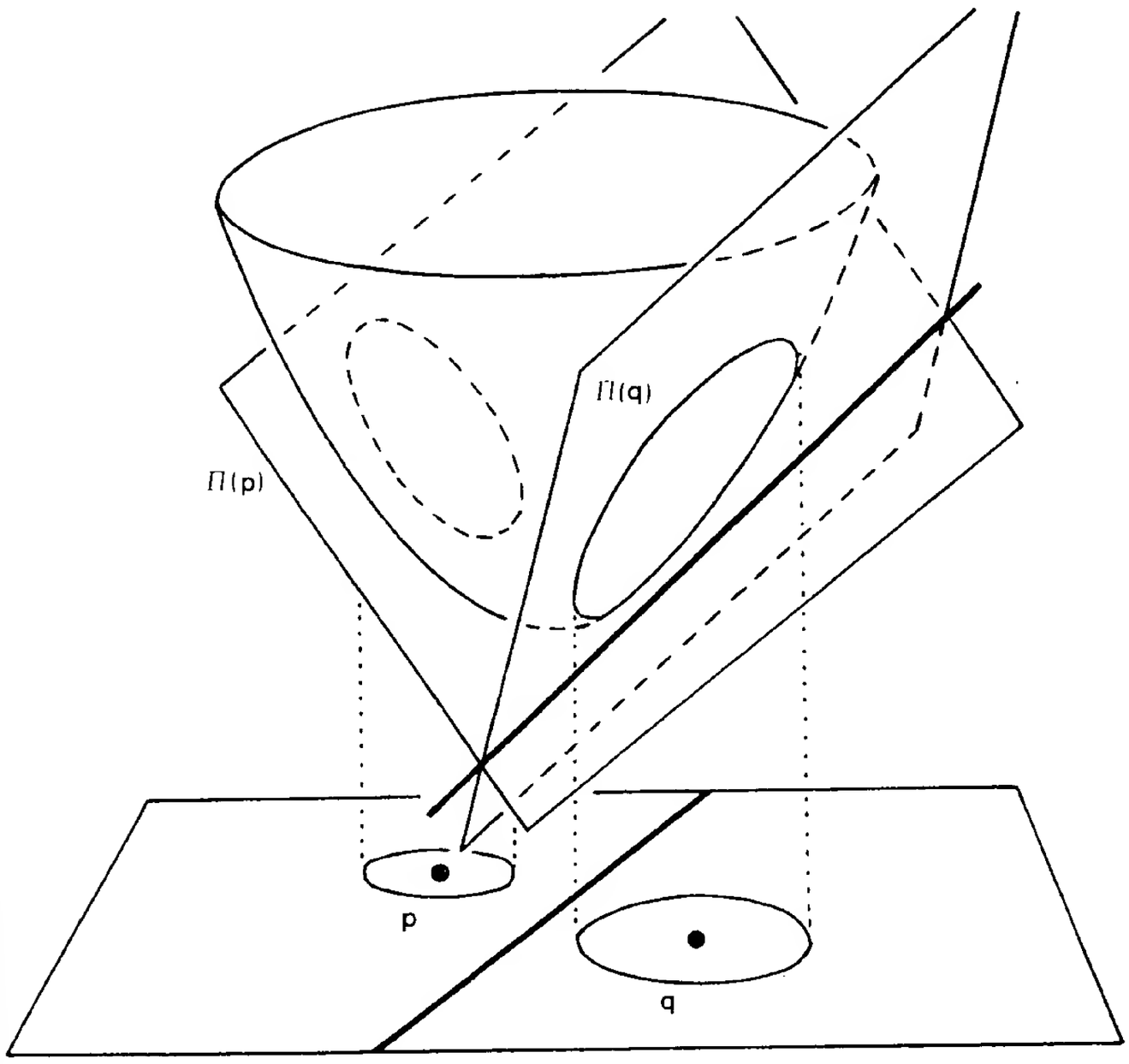


Figure 32. Spatial interpretation of chordale [Aurenhammer 1987a].

ts of less power with respect to p. *power cell* of p is given by

$$cell(p) = \bigcap_{q \in S - \{p\}} h(p, q).$$

ıalogy to classical Voronoi regions, *power diagram* of S, $PD(S)$ is the :x polyhedral complex defined in R^d ese cells; see Figure 4 for the case . $PD(S)$ coincides with the classical ıoi diagram of S in the case of ly weighted sites. Note that power may be empty if general weights ;ed.

technical reasons let us identify ith the linear subspace of R^{d+1} ;onal to the $(d+1)^{st}$ coordinate $_{d+1}$. The key observation for the ıce of objects in R^{d+1} that are re- o power diagrams is that the power ın pow(x, p) can be expressed by the hyperplane

$$\pi(p): x_{d+1} = 2x^T p - p^T p + w(p)$$

in R^{d+1} [Aurenhammer 1987a]. This fact is best explained by reference to Figure 32. Let p and q be two sites in R^2 whose weights are indicated by circles. Projecting the circles onto the paraboloid of revolution $x_3 = x_1^2 + x_2^2$ defines two planes $\pi(p)$ and $\pi(q)$ that intersect the paraboloid in these projections. It is an easy analytical exercise to prove that the line $\pi(p) \cap \pi(q)$ projects orthogonally to the chordale chor(p, q) in the $x_1 x_2$-plane. This correspondence holds for arbitrary dimensions. As an immediate consequence, $PD(S)$ is the orthogonal projection of the boundary of the polyhedron that comes from intersecting the halfspaces of R^{d+1} above $\pi(p)$, for all $p \in S$. In fact, π is a bijective mapping between

weighted sites in R^d and hyperplanes in R^{d+1}. Thus, for any intersection of upper half-spaces there exists a corresponding power diagram in one dimension lower and vice versa.

This general result has far-reaching implications. Most important, it shows that power diagrams are in a geometric and combinatorial sense equivalent to unbounded convex polyhedra or, more precisely, to their boundaries, that is, to *convex polyhedral surfaces*. Convex polyhedra are well-understood objects in discrete geometry, hence so are power diagrams. Exact upper and lower bounds on the numbers of their faces of various dimensions are known. In particular, $PD(S)$ in R^d realizes at most n cells, f_j j-dimensional faces ($1 \le j \le d-1$), and $f_0 - 1$ vertices, for

$$f_j = \sum_{i=0}^{a} \binom{i}{j}\binom{n-d+i-2}{i} + \sum_{i=0}^{b} \binom{d-i+1}{j}\binom{n-d+i-2}{i},$$

$$a = \left\lceil \frac{d}{2} \right\rceil, \quad b = \left\lfloor \frac{d}{2} \right\rfloor.$$

This follows from the so-called upper bound theorem [Brondsted 1983]. The numbers f_j are $O(n^{\lceil d/2 \rceil})$ for $0 \le j \le d-1$.

As for the facets of a convex polyhedron, the power cells are convex but possibly unbounded polyhedra. Not every half-space within a given set needs to contribute to a facet, so power cells may be empty or degenerate. $d+1$ or more hyperplanes in R^{d+1} intersect in a common point (unless parallelism occurs), so at least $d+1$ power cells meet at each vertex of $PD(S)$. In particular, there are no vertices if the cardinality n of S does not exceed d. These and many other properties of $PD(S)$ can be read off conveniently from its $(d+1)$-dimensional embedding.

3.1.2 Hyperplane Arrangements

In order to exploit the situation fully, we turn our attention to diagrams of higher order [Edelsbrunner 1987]; Aurenhammer [1987a]. To this end, the concept of a power cell is generalized to more than one site. Let T be a subset of k sites in S. The power cell of T is defined as

$$\mathrm{cell}(T) = \bigcap_{p \in T,\, q \in S-T} h(p,q).$$

The complex of all (nonempty) power cells arising from the $\binom{n}{k}$ subsets of S with fixed cardinality k is known as the *order-k power diagram* of S, k-$PD(S)$. Clearly 1-$PD(S)$ = $PD(S)$ holds. Note that k-$PD(S)$ is just the order-k Voronoi diagram of S provided all sites in S are equally weighted. $(n-1)$-$PD(S)$ is also called the *furthest site* power diagram of S.

Now let Z denote the intersection of the half-spaces below the hyperplanes $\pi(p)$ for all $p \in T$ and above the hyperplanes $\pi(q)$ for all $q \in S - T$. Due to the foregoing reasoning, Z projects orthogonally onto cell(T) in R^d. This observation draws the connection between order-k power diagrams and so-called *hyperplane arrangements*. The latter are cell complexes that arise from dissecting the space by a finite number of hyperplanes. Each cell in an arrangement is the intersection of some half-spaces below or above its defining n hyperplanes. If we let a *k-level* consist of all cells lying below k and above $n-k$ hyperplanes, we can conclude that the k-level induced by $\{\pi(p) \mid p \in S\}$ projects to the cells of k-$PD(S)$ for each k between 1 and $n-1$. This shows that each arrangement in R^{d+1} corresponds to a complete family of higher order power diagrams in R^d and vice versa. The correspondence between order-1 power diagrams and convex polyhedral surfaces discussed earlier is a special case of this situation.

Figure 33 depicts a cell (left) being the intersection of five upper and three lower half-spaces, that is, a three-level cell. Its orthogonal projection onto the x_1x_2-plane is an order-3 power cell among eight sites (right) whose boundary is indicated by bold lines. Note that the power cell is

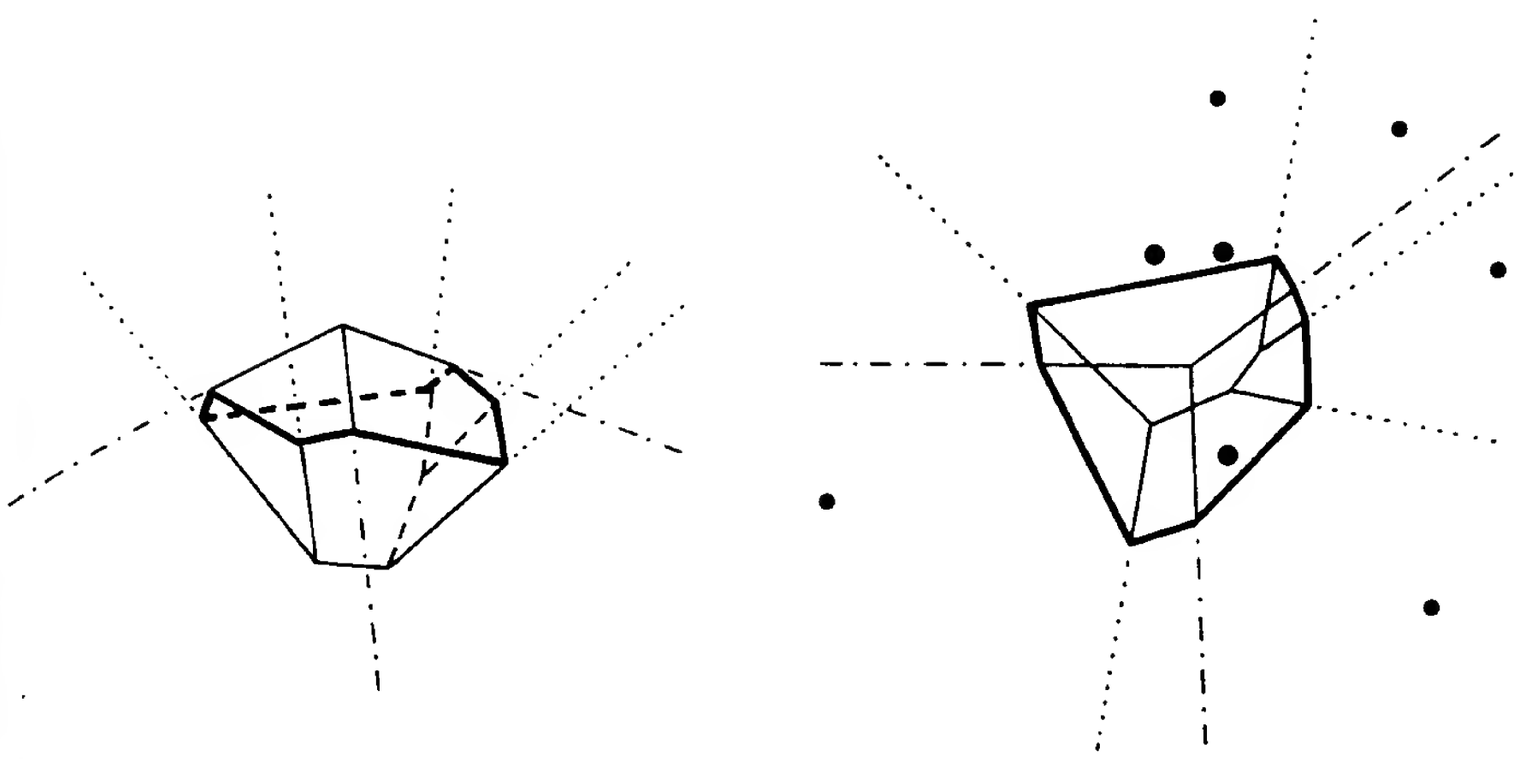


Figure 33. Projecting an arrangement cell [Aurenhammer 1987a].

carved into subcells by the furthest site power diagram of the three emphasized sites and by the power diagram of the remaining five sites.

The correspondence to arrangements straightforwardly yields an upper bound on the overall size of all higher order power diagrams of a fixed set of n sites: the maximal number of faces of an arrangement of n hyperplanes in R^{d+1}. This number is precisely

$$\sum_{i=d-j+1}^{d+1} \binom{i}{d-j+1} \binom{n}{i}$$

or faces of dimension j, which is in $)(n^{d+1})$ for $0 \le j \le d+1$ [Alexanderson nd Wetzel 1978]. Edelsbrunner et al. 1986a] developed an algorithm for contructing such arrangements in $O(n^{d+1})$ ime. Consequently, the complete family f higher order power (or Voronoi) diarams of a given set of sites can be omputed in optimal time and space. delsbrunner [1986] showed how to conruct the k-level of an arrangement R^3, and thus a single order-k ower (or Voronoi) diagram in the plane, a cost of $O(\sqrt{n} \log n)$ per edge. ecently, Mulmuley [1991] proposed a ndomized algorithm for constructing vels of order 1 to k in $d \ge 4$ dimenons in worst-case optimal time.

We state another interesting consequence. Let $\pi'(p)$ denote the image of reflection of the hyperplane $\pi(p)$ through R^d. The cells in the k-level of $\{\pi'(p) \mid p \in S\}$ correspond bijectively to the cells in the $(n-k)$-level of $\{\pi(p) \mid p \in S\}$ by reversing "upper" and "lower" for their supporting half-spaces. Clearly, any two corresponding cells yield the same orthogonal projection onto R^d. That is to say, they yield the same power cell that is simultaneously of order-k and order-$(n-k)$ for fixed k. This shows that any order-k power diagram for n sites can be intepreted as some order-$(n-k)$ power diagram. In particular, the furthest site power diagram of S is the order-1 power diagram of some set of sites (distinct from S, in general). We point out that this is not true for the subclass of classical Voronoi diagrams, as can be seen from the maximal size of their closest site and furthest site counterparts that differ even in R^2; see the Introduction and Section 2.2.1.

Finally, let us mention a result on power cells in R^2. Let $T \subset S$ and $|T| = k$. By definition, cell(T) is the intersection of $k(n-k)$ half planes and thus might have that many edges. On the other hand, cell(T) is the projection of a polyhedron in R^3 that has at most n facets and thus only $O(n)$ edges by

Euler's relation. It follows that cell(T) has $O(n)$ edges and can be constructed in $O(n \log n)$ time by intersecting n half-spaces of R^3.

3.1.3 Convex Hulls

The *convex hull* of a finite point set is the intersection of all half-spaces containing this set. Next, we study the duality of (order-1) power diagrams and convex hulls [Aurenhammer 1987a]. Duality is usually defined for convex polyhedra. Let Z and Z' be two convex polyhedra in R^{d+1}. Z and Z' are said to be *dual* if there is a bijective mapping ψ between the j-dimensional faces of Z and the $(d-j)$-dimensional faces of Z' such that $f \subseteq g$, for any two faces f and g of Z if and only if $\psi(g) \subseteq \psi(f)$. Since power diagrams in R^d are projections of convex polyhedral surfaces in R^{d+1}, the notion of duality carries over in a natural manner.

We now consider, for each site $p \in S$ with weight $w(p)$, a point in R^{d+1}:

$$\Delta(p) = \begin{pmatrix} p \\ p^T p - w(p) \end{pmatrix}.$$

From the formula expressing the hyperplane $\pi(p)$ it can be seen that $\Delta(p)$ and $\pi(p)$ are related via *polarity* with respect to the paraboloid of revolution. $\Delta(p)$ is the *pole* of $\pi(p)$ and $\pi(p)$ is the *polar hyperplane* of $\Delta(p)$. If $\pi(p)$ happens to intersect the paraboloid, then polarity has the geometric interpretation displayed in Figure 34. The hyperplanes touching the paraboloid at its points of intersection with $\pi(p)$ all concur at $\Delta(p)$.

Recall that the intersection, Z, of the half-spaces above $\pi(p)$, for all $p \in S$, projects to $PD(S)$. Take an arbitrary j-dimensional face of Z that lies in the intersection of, say, $\pi(p_1), \ldots, \pi(p_m)$, $m \geq d - j + 1$. That is, there exists some point on these but below all other hyperplanes defined by S. Since polarity preserves the relative position between points and hyperplanes, there is a hyperplane with $\Delta(p_1), \ldots, \Delta(p_m)$ on it and $\Delta(q)$ above it, for all remaining sites

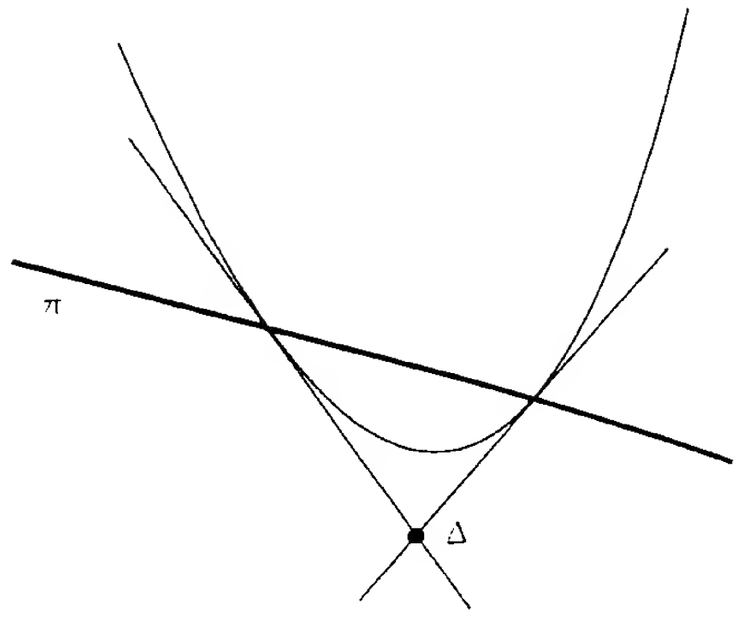


Figure 34. Polarity between points and lines.

$q \in S$. This, however, means that $\Delta(p_1), \ldots, \Delta(p_m)$ span a $(d-j)$-dimensional face of the *lower part* of the convex hull of the point-set $\{\Delta(p) \mid p \in S\}$. This lower part consists of all boundary faces of the hull that are visible from the point on the x_{d+1}-axis at $-\infty$. In other words, we have shown the duality of Z, and thus of $PD(S)$, to that lower convex hull part.

By the same reasoning, the duality of the furthest site power diagram of S and the *upper part* of the convex hull of $\{\Delta(p) \mid p \in S\}$ can be established. This reveals another interesting link between closest site and furthest site power diagrams and, particularly, classical Voronoi diagrams. Most important, these diagrams become constructable in general dimensions d via convex-hull algorithms. Running times of $O(n \log n)$ for $d = 2$ and of $O(n^{\lceil d/2 \rceil})$ for odd d are achieved, both of which are worst-case optimal; see Preparata and Hong [1977] and Seidel [1981], respectively. For even $d \geq 4$, these diagrams can be computed at logarithmic time per face [Seidel 1986].

By projecting down the edges of the lower and upper part of the convex hull that arises from a Voronoi diagram, one obtains the closest site and furthest site Delaunay triangulation, respectively. In such a triangulation, each edge is orthogonal to its dual facet. Conversely, the existence of some set of sites allowing a

triangulation whose edges are orthogonal to the facets of a given cell complex in R^d is a necessary and sufficient condition for that cell complex to be a power diagram or, equivalently, to be the projection of a convex polyhedral surface [Aurenhammer 1987b, 1987c]. If the given cell complex is *simple* (i.e., exactly $d + 1$ cells meet at each vertex), a projection surface can be reconstructed, if it exists, in time proportional to the number of facets of that cell complex. It is well known that such a surface always exists except for $d = 2$. This implies that any simple complex of n cells in $d \geq 3$ dimensions, which might have a large number of lower dimensional faces, can be specified by storing only n half-spaces in R^{d+1}.

Surprisingly, even higher order power diagrams are dual to some convex hull in R^{d+1} [Aurenhammer 1990a]. For any subset T of S with k sites let $\Delta(T)$ denote the point in R^{d+1} whose orthogonal projection onto R^d and whose $(d + 1)^{st}$ coordinate are

$$\sum_{p \in T} p \quad \text{and} \quad \sum_{p \in T} \left[p^T p - w(p) \right],$$

respectively. This definition conforms to the case $k = 1$ since $\Delta(T) = \Delta(p)$ when $T = \{p\}$. Using arguments similar as for $k = 1$, one may verify that the lower part of the convex hull of the point-set

$$S_k = \{\Delta(T) \mid T \subset S, |T| = k\}$$

is dual to k-$PD(S)$. As one corollary, the very lower part has to be dual to the intersection of some corresponding upper half-spaces and hence to an order-1 power diagram. To be more precise, k-$PD(S)$ coincides with $PD(S'_k)$, for S'_k being the (accordingly weighted) orthogonal projection of S_k onto R^d. This implies that the class of power diagrams is closed under order-k modification—another result that is not true for the more restrictive class of classical Voronoi diagrams. From the algorithmic viewpoint, k-$PD(S)$ becomes constructable via convex hull algorithms. For example, the planar Voronoi diagram of order-k can be obtained from its order-$(k - 1)$ predecessor in $O(kn \log n)$ time by simply computing a convex hull in R^3. Insertions and deletions of sites in an order-k Voronoi diagram also amount to computing certain convex hulls; these operations even can be handled on-line in an efficient manner [Aurenhammer and Schwarzkopf 1991].

3.1.4 *k*-Sets

Let Q be a set of n points in R^{d+1}. A *k-set* of Q is a subset of k points of Q that can be separated from the remaining $n - k$ points by some hyperplane of R^{d+1}. The discussion in the foregoing section reveals a correspondence between k-sets and order-k power diagrams.

By polarity, points in Q can be transformed into hyperplanes; each point $q \in Q$ bijectively corresponds to a weighted site p in R^d such that $\Delta(p) = q$ and thus to the hyperplane $\pi(p)$ in R^{d+1}. Now, let M be a k-set of Q, and consider a hyperplane h that separates M from $Q - M$. Without loss of generality, M lies below h. Equivalently, the pole of h lies below all polar hyperplanes of the points in M and above all polar hyperplanes of the points in $Q - M$. Hence the intersection of the corresponding lower and upper half-spaces of R^{d+1} bounded by these polar hyperplanes is nonempty. This intersection, being a cell in the k-level of a hyperplane arrangement, projects to a cell of the order-k power diagram of a set of weighted sites, $\{p \in R^d \mid \Delta(p) \in Q\}$.

We have obtained a one-to-one correspondence between k-sets in R^{d+1} and power cells in R^d. This allows us to apply known results on the number of k-sets to the analysis of the size of order-k power diagrams. Let $f_d(k, n)$ denote the maximum number of k-sets of any set of n points in R^d. Known asymptotic values of this function are $f_2(k, n) = \Omega(n \log k)$ and $f_2(k, n) = O(n\sqrt{k})$ [Edelsbrunner and Welzl 1985], $f_3(k, n) = \Omega(nk \log k)$ and $f_3(k, n) = O(n^{8/3} \log^{5/3} n)$ [Edelsbrunner et al. 1986; Aronov et al. 1990, respectively]. Not surprisingly, a

better understood quantity is

$$g_d(k, n) = \sum_{j=1}^{k} f_d(j, n).$$

Chazelle and Preparata [1986] showed $g_3(k, n) = O(nk^5)$ and Clarkson [1987] showed $g_3(k, n) = O(n^{1+\epsilon}k^2)$, for any $\epsilon > 0$. Note that $g_d(n-1, n) = O(n^d)$ according to the maximum number of cells of a hyperplane arrangement in R^d. Recently, Clarkson and Schor [1989] succeeded in proving $g_d(k, n) = \Theta(n^{\lfloor d/2 \rfloor} k^{\lceil d/2 \rceil})$ as $n/k \to \infty$ for fixed d. Determining exact or at least asymptotically tight bounds on $f_d(k, n)$ still remains an important open problem.

3.1.5 Related Diagrams

The central role of power diagrams within the context of Voronoi diagrams becomes even more apparent by the observations described below [Aurenhammer and Imai 1988]. Let p, q, and r denote three weighted sites. For the three chordales they define, $\text{chor}(p, q) \cap \text{chor}(q, r) \subset \text{chor}(p, r)$ necessarily holds. It is not difficult to see that this condition is also sufficient for three hyperplanes to be the chordales defined by three sites if we keep in mind the relationship between chordales in R^d and intersections of hyperplanes in R^{d+1}. On the other hand, the condition is trivially fulfilled by the separators for any Voronoi diagram with polyhedral regions since the regions would not form a cell complex otherwise. We thus conclude that any Voronoi diagram whose separators are hyperplanes is a power diagram. This result applies to diagrams defined by the *general quadratic-form distance*

$$Q(x, p) = (x - p)^T M(x - p) - w(p),$$

with M a nonsingular and symmetric $(d \times d)$-matrix and thus to many particular distance functions considered in the literature. Note that Q equals the power function if M is the identity matrix.

One is tempted to believe that power diagrams are related solely to objects in higher dimensions. Below we will show that a certain rather general type of Voronoi diagram in R^{d-1} can be embedded into a power diagram in R^d. To this end, we identify R^{d-1} with the linear subspace $x_d = 0$ of R^d. Let p be some point site in R^{d-1} and consider an arbitrary distance function $f(x, p)$ for p. For some strictly increasing function F on R, we define the *cone* of p with respect to F as

$$\text{cone}_F(p) = \left\{ \begin{pmatrix} x \\ x_d \end{pmatrix} \mid x \in R^{d-1},\ x_d = F(f(x, p)) \right\}.$$

A Voronoi diagram for point sites in R^{d-1} under the distance function f is termed *transformable* if there exists an F that forces the affine hull, $\alpha(p, q)$, of $\text{cone}_F(p) \cap \text{cone}_F(q)$ to be a hyperplane in R^d for any distinct sites p and q. Transformable diagrams represent projected sections of power diagrams in R^d. A point $x \in R^{d-1}$ satisfies $f(x, p) < f(x, q)$ if and only if its projection, x', onto $\text{cone}_F(p)$ lies in a fixed (open) halfspace, $h(p, q)$, bounded by $\alpha(p, q)$. As a consequence, x belongs to p's region exactly if

$$x' \in Z = \bigcap_{q \in S - \{p\}} h(p, q),$$

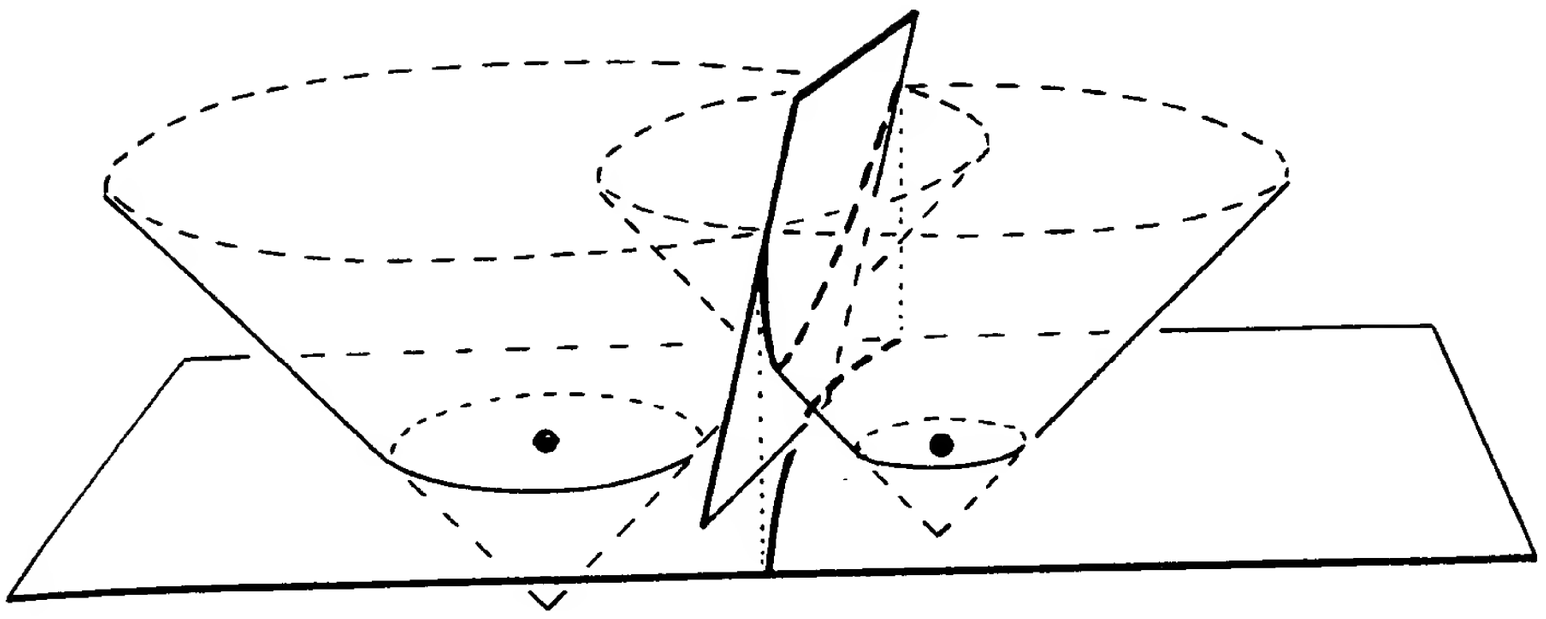
for S being the underlying set of sites in R^{d-1}. But the d-dimensional polyhedron Z is a power cell since $\alpha(p, q) \cap \alpha(q, r) \subset \alpha(p, r)$ holds so that these affine hulls can be interpreted as chordales. In conclusion, the boundary of p's region is the orthogonal projection onto R^{d-1} of $\text{cone}_F(p)$ intersected with a power cell in R^d.

Transformable Voronoi diagrams occur for the *additively weighted distance*,

$$a(x, p) = \delta(x, p) - w(p) \quad \text{with } F(a) = a,$$

and the *multiplicatively weighted distance*,

$$m(x, p) = \frac{\delta(x, p)}{w(p)} \quad \text{with } F(m) = 2m^2.$$



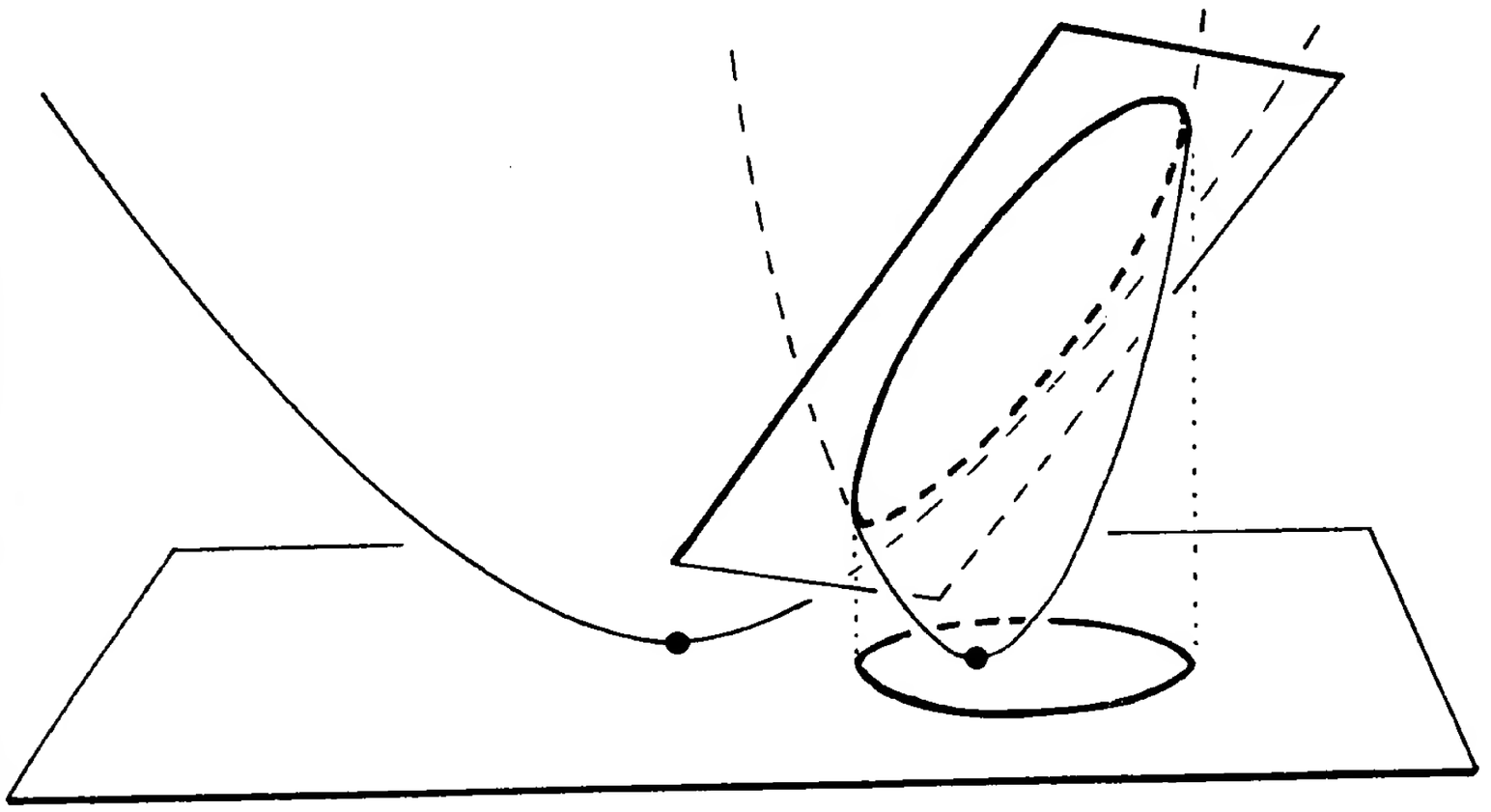


Figure 35. Cones for two distance functions.

The former gives rise to hyperbolically bounded and star-shaped regions (the Johnson-Mehl model, Figure 6), whereas the latter induces spherically bounded regions (the Apollonius model, Figure 12). Its regions are disconnected, in general, which already reflects the major complexity of the latter type.

Figure 35 displays cones for the additively weighted distance (above) and the multiplicatively weighted distance (below). They are cones of revolution in the former case—intersecting in a hyperbola that projects down to a hyperbolic separator—and paraboloids of revolution in the latter case—intersecting in an ellipsis that projects down to a circular separator. In both cases, the affine hull of cone intersections is a plane.

Upper bounds on the size of both types of diagram in d dimensions can be derived from the maximum size of a power diagram in R^{d+1}. Concerning the algorithmic aspects of this relationship, the only known algorithms for constructing these diagrams in $d \geq 3$ dimensions are obtained. The Apollonius model in the plane can be computed in $O(n^2)$ time and

space, which is worst-case optimal. We mention that the notion of embeddability generalizes nicely to order k.

3.1.6 Upper Envelopes

Let $f_1, \ldots, f_t$ be piecewise linear d-variate functions on R^d. In the following, we will not distinguish between a function on R^d and its graph in R^{d+1}. Consider the pointwise maximum, U, of these functions:

$$U(x) = \max_{1 \le i \le t} f_i(x).$$

U is called the *upper envelope* of $f_1, \ldots, f_t$. We have already observed that certain upper envelopes are related to power diagrams. For any set S of weighted sites in R^d, the upper envelope of the hyperplanes $\pi(p)$, $p \in S$, projects to $PD(S)$. Below we briefly consider two types of Voronoi diagrams where this relationship generalizes particularly well.

Let S be a set of n point sites in the plane, and consider a partition of S into subsets $C_1, \ldots, C_t$ called *clusters*. The *Hausdorff distance* of a point x to the cluster C_i is defined as

$$h(x, C_i) = \max\{\delta(x, p) \mid p \in C_i\}.$$

h is also called the complete linkage distance; compare Section 2.5.1. Each cluster C_i may be associated with a convex polyhedral surface $\mathrm{cap}(C_i)$ in R^3, being the boundary of the polyhedron that comes from intersecting the half-spaces below $\pi(p)$ for all $p \in C_i$. Easy arguments show the following. For any point x in the plane, $h(x, C_i) < h(x, C_j)$ if and only if the vertical line through x intersects $\mathrm{cap}(C_i)$ in a point lying above $\mathrm{cap}(C_j)$. This implies that the upper envelope of $\mathrm{cap}(C_1), \ldots, \mathrm{cap}(C_t)$ projects to the Voronoi diagram induced by $C_1, \ldots, C_t$ and h. For an illustration of the Voronoi diagram for two-site clusters see Figure 30.[13]

To obtain another example, assume that visibility of the sites is constrained to a *window*, W, a segment on a line avoiding the convex hull of S. A site $p \in S$ and a point x are called *visible* if the line segment joining x and p intersects W. The *peeper's Voronoi diagram* of S and W assigns each point in the plane to the region of the closest site visible from it. See Figure 36 where the window is shown between two bold line segments. The diagram is composed of perpendicular bisectors of sites (dashed) and of rays through sites and window endpoints (solid).

Each site p may be associated with an unbounded convex polygon $\mathrm{plate}(p)$ in R^3, being the projection onto $\pi(p)$ of the set of all points visible from p. Again it is easy to see that a point x falls into the region of a site p just if the vertical line through x intersects $\mathrm{plate}(p)$ in a point that lies above $\mathrm{plate}(q)$ for all $q \in S - \{p\}$. We conclude that the upper envelope of $\mathrm{plate}(p)$, $p \in S$, projects to the peeper's Voronoi diagram of S.

Upper envelopes of piecewise linear functions on R^2 are well-studied geometric objects. Pach and Sharir [1989] showed that they can attain a size of $\Theta(n^2\alpha(n))$, where n is the total number of triangles needed in partitioning the linear pieces of these functions and where α denotes the inverse of Ackermann's function. Construction algorithms exist that run in time proportional to their worst-case size; see Edelsbrunner et al. [1989] who also give a lower bound of $\Omega(n^2)$ on the size of a cluster Voronoi diagram. Recent results on upper envelopes by Huttenlocher et al. [1991] imply that the diagram for t clusters with a total of n sites has a size of $O(tn\alpha(tn))$. The upper envelope arising from the peeper's Voronoi diagram has a worst case size of $\Theta(n^2)$ [Aurenhammer and Stöckl 1988]. Note that the superlinear size of a diagram implies the existence of disconnected regions.

3.2 Topology of Planar Diagrams: Divide-and-Conquer Construction and Its Variants

By far not all types of Voronoi diagrams considered in the literature can be

[13] Observe that the upper envelope of $\mathrm{cap}(T)$, for all subsets T of k sites of S, consists of faces of the k-level of $\{\pi(p) \mid p \in S\}$ and thus projects to the order-k Voronoi diagram of S.

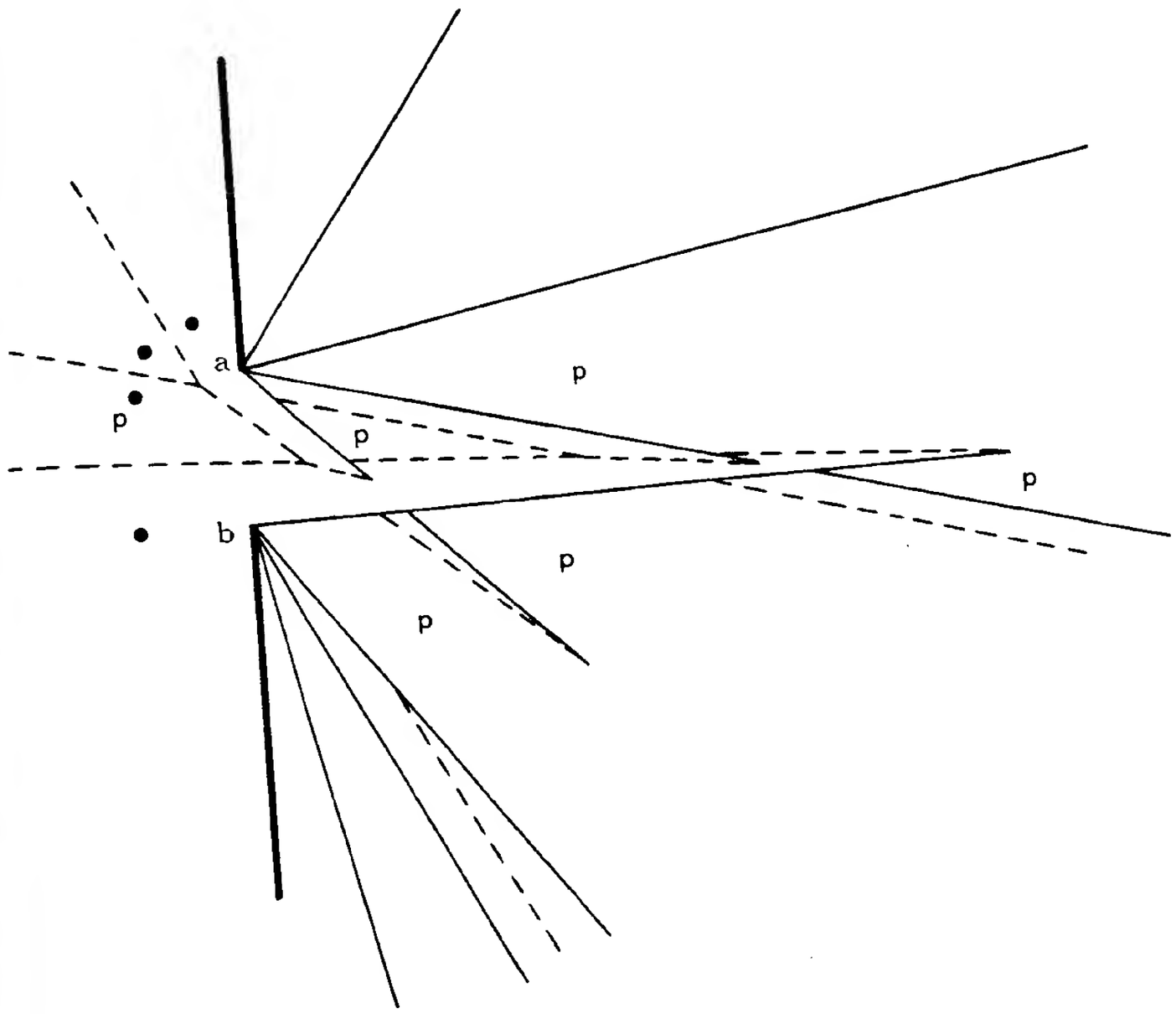


Figure 36. Peeper's Voronoi diagram.

ght into connection with geometric
ts having nice algorithmic proper-
like convex hulls or hyperplane ar-
ements. So there is need for finding
res common to such "ungeometric"
ams. We do this by extracting topo-
l properties, being guided by their
ance to the divide-and-conquer con-
ion of Voronoi diagrams. Attention
tricted to the plane as the instance
st applications and easiest analysis.
concept of diagram we are inter-
in here may be characterized as
s. The n sites are points in R^2.
istance function varies by different
s of its "circles" that may depend
al properties of R^2. It thus may
or the individual sites.

Convex Distance Functions

portant type fitting into the con-
bove is generated by the *convex distance function* or the *Minkowski distance* [Chew and Drysdale 1985]. Let C denote the boundary of some compact and convex subset of R^2 with the origin o in its interior. The distance with respect to C of o to some point $x \in R^2$ is given by

$$d_C(x, o) = \frac{\delta(x, o)}{\delta(x', o)}$$

for x' being the point of intersection of C and the ray from o to x. Clearly d_C can be defined with respect to any site q of a given set S by translating C so that q occupies the former position of o. Observe that $d_C(x, q)$ depends on the euclidean distance $\delta(x, q)$ as well as on the direction of the ray from q to x. Convex distance functions include, among others, the general L_p*-metric*

$$\delta_p(x, q) = \sqrt[p]{|q_1 - x_1|^p + |q_2 - x_2|^p},$$

with

$$q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \quad \text{and} \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},$$

whose most important instances are for $p = 1$ (the *Manhattan metric*; C is a square rotated by 45°), $p = 2$ (the *euclidean distance*; C is a circle), and $p = \infty$ (the *Maximum metric*; C is an axis-parallel square). Note that d_C is a metric only if C is point-symmetric with respect to q since $d_C(q, r)$ might differ from $d_C(r, q)$ otherwise. However, the triangle inequality

$$d_C(q, r) + d_C(r, s) \geq d_C(q, s)$$

can easily be shown to hold even if C is not point-symmetric.

Let us now investigate the Voronoi diagram defined by d_C and a set S of n sites in R^2. From the classical type we adopt the notions of separator $\text{sep}(q, r)$ of q and r, dominance $\text{dom}(q, r)$ of q over r, and region $\text{reg}(q)$ of q; compare the Introduction. Clearly $\text{sep}(q, r)$ need not be a straight line as is the case for euclidean distance. An even more unpleasant phenomenon is that $\text{sep}(q, r)$ fails to be one dimensional in general. To remedy this shortcoming, the notions above may be redefined slightly by referring to the lexicographical order (symbolized by $<$) of the sites in S [Klein and Wood 1988]. The dominance of q over r is redefined as

$$\text{Dom}(q, r) = \begin{cases} \text{dom}(q, r) - \text{sep}(q, r), & \text{if } q < r \\ \text{dom}(q, r), & \text{otherwise.} \end{cases}$$

The new separator, $\text{Sep}(q, r)$, is the intersection of the boundaries of $\text{Dom}(q, r)$ and $\text{Dom}(r, q)$, which is obviously one-dimensional. Figure 37 shows a two-dimensional separator in the L_1-metric (left) and its redefined version (right). For a Voronoi diagram in the L_1-metric, see Figure 3. The region of a site q now can be rewritten as

$$\text{Reg}(q) = \bigcap_{r \in S - \{q\}} \text{Dom}(q, r).$$

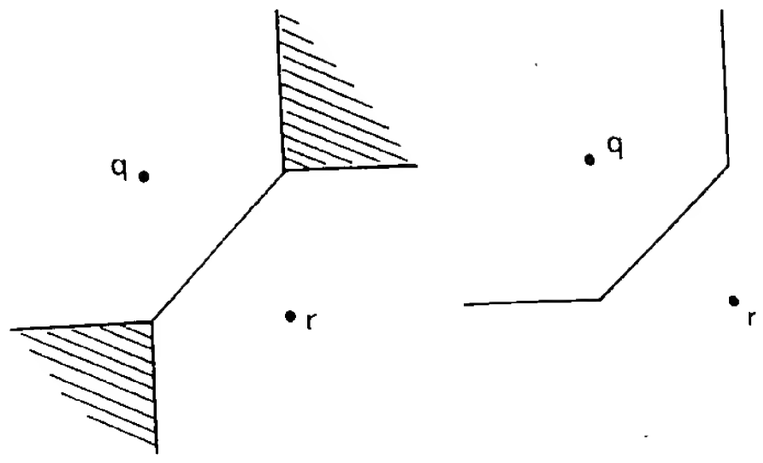


Figure 37. Redefining a separator.

That is, each point in R^2 belongs to the region of the lexicographically least site among its closest sites in S. This ensures that the interiors of the regions cover the plane up to a set of Lebesgue measure zero.

Defined in this way, the Voronoi diagram, $V_C(S)$, for a convex distance function d_C has the nice property that its regions are *star shaped*; $\text{Reg}(q)$ contains the straight-line segment between q and x for each $x \in \text{Reg}(q)$. To prove this by contradiction, assume the existence of a point y on this segment with $d_C(y, q) > d_C(q, r)$, for $r \in S - \{q\}$. This implies $d_C(x, q) = d_C(x, y) + d_C(y, q) > d_C(x, y) + d_C(y, r)$. By the triangle inequality, the latter sum is greater than or equal to $d_C(x, r)$ so that $d_C(x, q) > d_C(x, r)$, that is, $x \notin \text{Reg}(q)$, follows.

The star shapedness of the regions implies their simple connectedness. Clearly, regions cannot vanish since $q \in \text{Reg}(q)$ holds for all $q \in S$. So $V_C(S)$ can be viewed as a planar graph with exactly n regions. Since vertex degrees are at least three, there are $O(n)$ edges and vertices. We conclude a linear bound on the size of $V_C(S)$.

3.2.2 *Divide-and-Conquer Construction*

Our next aim is to show that Voronoi diagrams for convex distance functions can be constructed efficiently by a *divide-and-conquer* algorithm. Divide and conquer splits the problem at hand into two smaller subproblems, computes their solutions recursively (unless they

are very small and can be solved by trivial methods), and finally combines the partial solutions to the global one. The V_C type of diagram is well suited to attack by this strategy. The underlying set S of sites is partitioned into two subsets S_1 and S_2 of nearly equal cardinality. $V_C(S_1)$ and $V_C(S_2)$ are computed recursively and are then merged to $V_C(S)$.

The merge step deserves the main attention since it actually constructs the diagram. We will discuss this process in some detail and under the simplifying but computationally unrestrictive assumption that S_1 and S_2 are separable by a vertical line. Since $V_C(S_1)$ as well as $V_C(S_2)$ defines a partition of R^2, any point $x \in R^2$ falls into the region $\mathrm{Reg}_1(q)$ of $V_C(S_1)$ for some site $q \in S_1$ and into the region $\mathrm{Reg}_2(r)$ of $V_C(S_2)$ for some site $r \in S_2$. Now observe that $x \in \mathrm{Reg}(q)$ of $V_C(S)$ if $d_C(x, q) < d_C(x, r)$, and $x \in \mathrm{Reg}(r)$ of $V_C(S)$ otherwise. So we have to cut off some part from $\mathrm{Reg}_1(q)$ and from $\mathrm{Reg}_2(r)$ by means of $\mathrm{Sep}(q, r)$ in order to obtain $\mathrm{Reg}(q)$ and $\mathrm{Reg}(r)$. Carrying out this task for all relevant pairs of sites constitutes the merge process. The union of the newly integrated pieces of separators will be called the *merge chain*, $M(S_1, S_2)$.

The efficiency of an implementation of the merge process critically depends on the topological properties of the merge chain. We first show that $M(S_1, S_2)$ behaves well if the euclidean distance δ is taken for d_C [Shamos and Hoey 1975]. As a matter of fact, each horizontal line intersects $M(S_1, S_2)$ in exactly one point in this case, that is to say, $M(S_1, S_2)$ is (vertically) *monotone*. $M(S_1, S_2)$ is a polygonal line composed of edges of the classical diagram $V(S)$ that are portions of perpendicular bisectors of sites in S_1 and S_2, respectively. Figure 17 gives an illustration. If there were a horizontal line intersecting the merge chain in two or more points, some site $q \in S_1$ would be forced to have a larger x_1-coordinate than a site $r \in S_2$. This contradicts the assumption that S_1 and S_2 are separable by a vertical line; compare Figure 38.

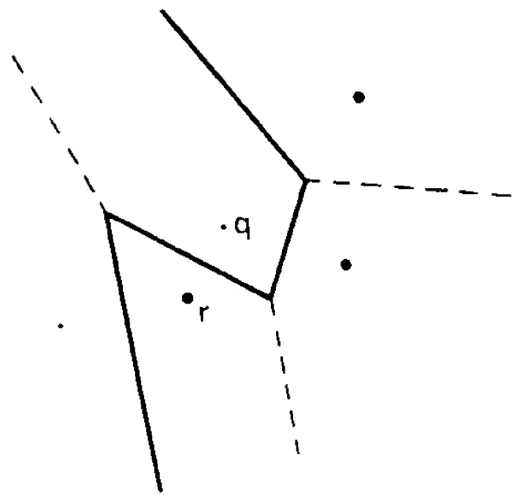


Figure 38. Impossible merge chain.

The monotonicity of $M(S_1, S_2)$ implies its connectedness and unboundedness. Hence the merge chain may be constructed edge by edge, starting with an unbounded one. As is mentioned in the Introduction, unbounded edges separate sites on the convex hull of S. So it suffices to determine a site in S_1 and a site in S_2 that are neighbors on the boundary of the convex hull of S. This can be accomplished in $O(n)$ time by finding a line tangent to the convex hulls of respective S_1 and S_2. Each further edge of $M(S_1, S_2)$ is found by tracing the boundaries of the current regions of $V(S_1)$ and $V(S_2)$ in an appropriate direction until they intersect the actual separator. Easy counting arguments show that only $O(n)$ edges are processed in total: Each edge traced is either discarded or shortened to a new edge of $V(S)$. This implies an overall runtime of $O(n)$ for the merge process. By the recurrence relation $T(n) = 2T(n/2) + O(n)$ that results from using divide and conquer, the time complexity $T(n)$ for computing a classical Voronoi diagram is $O(n \log n)$.

Let us now come back to the distance d_C for general convex shapes C [Chew and Drysdale 1985]. As in the euclidean case, $M(S_1, S_2)$ is composed of those parts of the boundaries of regions of $V_C(S)$ that separate sites in S_1 from sites in S_2. Thus $M(S_1, S_2)$ is one dimensional. Potentially, this merge chain may consist of cyclic and acyclic (but then necessarily unbounded) topological curves in R^2.

Examples show that $M(S_1, S_2)$ is in fact disconnected in general. Its connected components all, however, are acyclic and thus unbounded. Let us prove that there are no cycles by assuming the existence of some cycle M and deducing a contradiction. Without loss of generality, M encircles a subset T of sites in S_1. Clearly, then M does not encircle any site in S_2. Furthermore, there exists a leftmost point $x \in M$ with $x \in \mathrm{Sep}(r, s)$ for some $r, s \in S_2$. Since the regions of $V_C(S)$ are star shaped, the three open line segments from x to r, s, and some $q \in T$, respectively, do not intersect M. Thus, as r and s lie to the right of q, x has to lie to the left of q. This, however, contradicts the convexity of the underlying shape. C encircles x and passes through q, r, and s since the last three points are equidistant from x with respect to d_C. Figure 39 illustrates this situation; M and C are shown bold and dashed, respectively.

In order to construct the acyclic merge chain $M(S_1, S_2)$, one needs to detect some unbounded edge of each of its connected components. The previously described procedure of tracing boundaries of regions may then be applied. Unfortunately, the convex-hull method that works successfully in the euclidean case fails to be correct for d_C. To get an alternative method, we observe that each unbounded edge has to be contained in the intersection of two unbounded regions of $V_C(S_1)$ and $V_C(S_2)$, respectively. By a circular scan through these regions, each relevant pair $\mathrm{Reg}_1(q)$ and $\mathrm{Reg}_2(r)$ can be found and intersected with $\mathrm{Sep}(q, r)$ in a total of $O(n)$ time. If $\mathrm{Reg}_1(q) \cap \mathrm{Reg}_2(r) \cap \mathrm{Sep}(q, r)$ extends to infinity, then this edge is taken as the starting edge for a connected component of $M(S_1, S_2)$.

The linear-time behavior of the merge process thus can be maintained for general convex distance functions d_C. Consequently, $V_C(S)$ requires $O(n \log n)$ time and $O(n)$ space for construction. Of course, the complexity analysis is based on the assumption that d_C is *computationally simple*. That is, $d_C(q, x)$ can be calculated in $O(1)$ time for all $q \in S$ and $x \in R^2$, and, in addition, its separators can be computed and intersected in $O(1)$ time. This may not be the case for complex shapes C, however.

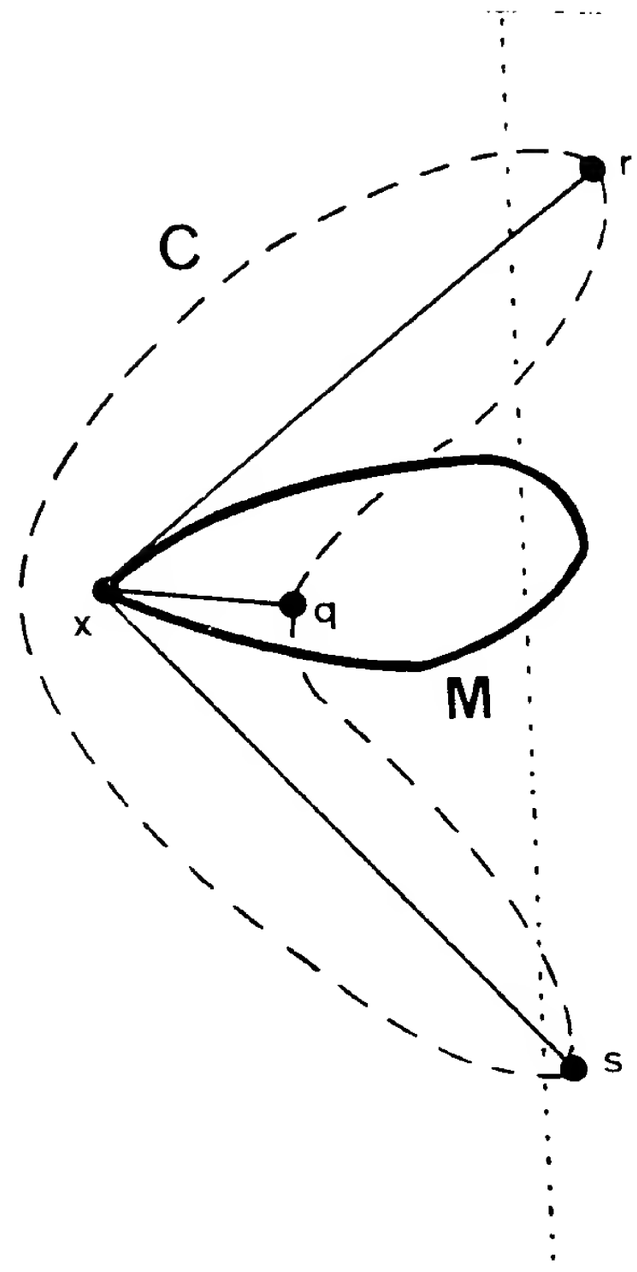


Figure 39. Cycles do not occur.

3.2.3 Nice Metrics

We have seen that divide and conquer is a powerful approach to the construction of generalized Voronoi diagrams. It also applies well to sites more general than points; see Section 1.3.3. We do not pursue these modifications here, however, since they do not fit into the present concept of a diagram. Instead, we attempt to characterize metrics in R^2 that preserve the two main properties of a Voronoi diagram exploited in the preceeding discussion [Klein 1989; Klein and Wood 1988]: the connectedness of regions

that implies an $O(n)$ size and the absence of cycles in the merge curve that allows an $O(n \log n)$ time construction.

Let d be any metric in R^2 that induces the euclidean topology. A curve γ in R^2 is termed *d-straight* if, for any three consecutive points x, y, z on γ, the equality

$$d(x, y) + d(y, z) = d(x, z)$$

olds. Clearly δ-straight curves are straight-line segments, for δ being the uclidean metric. By arguments similar o that used for convex distance functions namely using the triangle inequality), it an be verified that any Voronoi region eg(p) for d is *d-star-shaped*: $\gamma \subset \mathrm{Reg}(p)$ olds for the d-straight curve γ from p any $x \in \mathrm{Reg}(p)$. The d-star-shapedness Reg(p) does not necessarily imply its nnectedness since d-straight curves do t always exist. If, however, for any two ints $x, z \in R^2$ there is some y distinct m x, z with $d(x, y) + d(y, z) = x, z$), then the existence of γ is guarteed. Hence the Voronoi regions under se metrics are connected.

Posing an additional restriction on the tric d makes the Voronoi diagram n more well behaved: Assume that, any point $m \in R^2$ and for any positive mber r, the generalized disk

$$D = \{x \in R^2 \mid d(x, m) \leq r\}$$

h center m and radius r under the ric d is a simply connected set. For d r, the shape of D may vary with position of m provided D remains ply connected. *Symmetric* convex distce functions d_C (i.e., with point-metric shape C) constitute a special of translation-invariant disks. And, ed, the proof that merge chains are lic can be extended from convex dise functions to d.

underline the usefulness of this conlet us give some examples for d. Let any point in R^2. The *Moscow met*ith respect to K induces d-straight s composed of pieces straight to-K or radially around K of minimal euclidean length. Figure 40 depicts a d-straight curve (dashed) between two points x and y and a Voronoi diagram (solid) for eight sites in this metric. For the *geodesic distance* among polygonal obstacles, d-straight curves are the minimum euclidean length connections avoiding any edge of the polygons (compare Figure 26). Also of interest are *composite metrics* where, for instance, distances are measured in the L_1-metric in some portion of R^2 and in the L_2- (i.e., euclidean) metric in the complement. Here, d-straight curves are composed of those in L_1 and L_2 such that curve length is minimized. In all three examples, the respective disks are simply connected. As a consequence, the resulting Voronoi diagrams—except for the geodesic distance function that fails to be computationally simple—can be computed in optimal $O(n \log n)$ time and $O(n)$ space by divide and conquer. Thereby, the underlying set of point sites has to be divided into subsets by curves whose intersections with any disk for d is either connected or empty. Consult Klein [1989] for further details. We pose it as an open question to find equivalent characterizations for metrics whose Voronoi diagrams are constructable by divide and conquer in optimal time.

3.3 Deformation of the Voronoi Diagram: Plane-Sweep Technique

The variety of techniques that can be used for constructing Voronoi diagrams is possibly as fascinating as the diagrams themselves. The construction of the classical planar Voronoi diagram by means of divide and conquer or via three-dimensional convex hulls are well-investigated algorithmic problems. Although the achieved complexity bounds are worst-case optimal, researchers continued considering alternative methods of construction. Aside from the incremental insertion strategy described in Sections 1.3.1 and 1.3.2, the plane-sweep technique to be discussed now stands out by its conceptual and computational simplicity. It extends nicely to various generalized types of Voronoi diagrams.

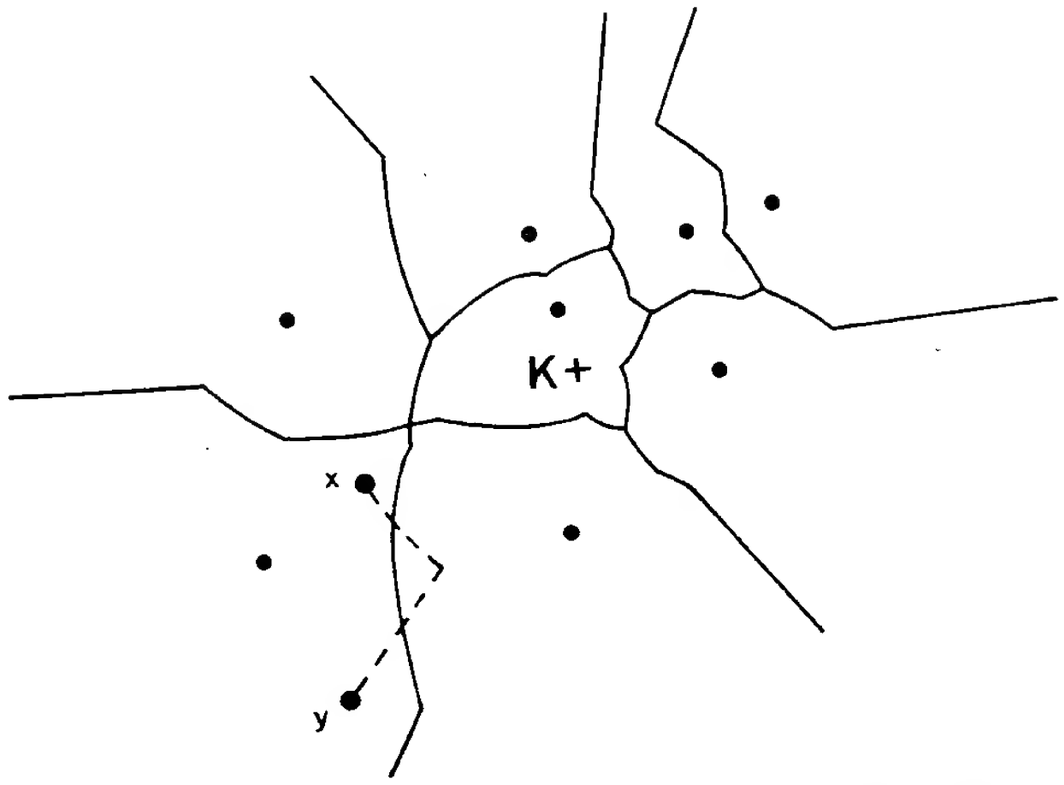


Figure 40. Voronoi diagram in the Moscow metric [Klein 1989].

3.3.1 *Deformation*

Let S denote a set of n point sites in R^2 and let $V(S)$ be their Voronoi diagram. As we will see, there are serious reasons for refraining from a direct construction of $V(S)$ via the plane-sweep technique. Therefore, first we introduce a continuous deformation of $V(S)$ [Fortune 1987]. Then, we describe the plane-sweep technique and its use for computing this deformation.

To explain the mechanism of deformation, the euclidean distance function with respect to the sites is interpreted in the following way. With each site $p \in S$, a cone

$$\text{cone}(p) = \{(x, z) \in R^3 \mid x \in R^2,\ z = \delta(x, p)\}$$

is associated. Cone(p) is an upwardly directed cone in R^3 with vertical axis of revolution, with apex p, and with an interior angle of $\pi/2$.[14] Cones may be viewed as bivariate functions on R^2. We define the *lower envelope* of $\{\text{cone}(p) \mid p \in S\}$ as the pointwise minimum of these functions or, equivalently, as the surface composed of that portion of each cone that lies below all other cones. From the definition of cone(p) it is evident that this lower envelope projects vertically to $V(S)$ onto R^2.

For our purposes it is preferable to project the envelope onto R^2 in a different way, namely under an angle of $\pi/4$ in positive x_2-direction. This already gives the desired deformation $V^*(S)$ of $V(S)$. To be more precise, the deformation * maps each point $x \in \text{reg}(p)$, say

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \qquad \text{into } x^* = \begin{pmatrix} x_1 \\ x_2 + \delta(x, p) \end{pmatrix}.$$

Let us study what happens in this process. Clearly, sites are invariant under *. Each separator sep(p, q) is deformed into a hyperbola with bottommost point p if p is below q and bottommost point q, otherwise. (This hyperbola degenerates to a vertical half line if p and q have the same x_2-coordinate.) Consequently, the deformed region reg*(p) of p is the intersection of hyperbolically bounded half planes of R^2. Reference to the upper envelope model shows that * preserves the topological properties of a Voronoi diagram. For instance, the interiors of the

[14] Note that the definition of cone(p) agrees with that of cone$_F$(p) used for transformable Voronoi diagrams in Section 3.1.5; F is the identity function.

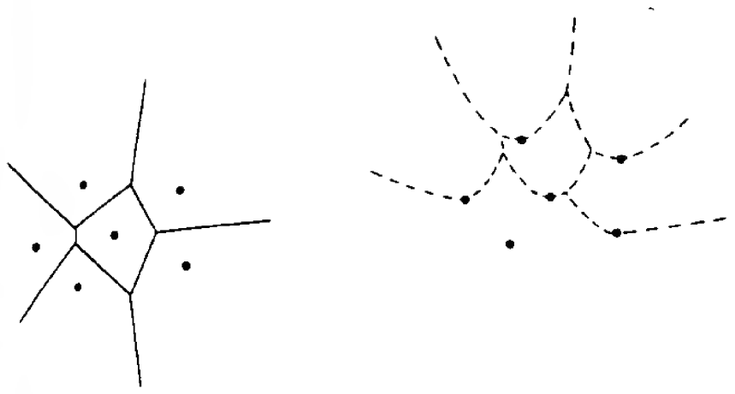


Figure 41. Deforming a Voronoi diagram [Fortune 1987].

deformed regions cover R^2 up to a set of Lebesgue measure zero. The algorithmic advantage of the deformation over the original is that the bottommost point of each region is its defining site. Exceptions are regions for the bottommost sites in S that necessarily are unbounded. Figure 41 shows the Voronoi diagram of six sites (left) and its deformation (right).

3.3.2 *Plane-Sweep Technique*

The properties of $V^*(S)$ are now exploited for its construction using the *plane-sweep technique*. Generally, this technique proceeds as follows. A horizontal line L is swept across the object to be constructed from below by keeping the invariant that the portion of the object below L is complete at any point in time. During the plane sweep, the cross section of L with this object has to be updated at certain critical points. We thus have to handle a one-dimensional dynamic problem instead of a two-dimensional static problem.

The applicability of the plane-sweep technique crucially depends on whether the "critical points" can be predicted. This is, for example, not the case if the Voronoi diagram of S is to be constructed in its original: A site $p \in S$ might lie above L while its region reg(p) already extends to below L. And it is important to know when regions begin to intersect L in order to integrate them properly into the cross section. We do not run into such problems when constructing $V^*(S)$, however, since the bottommost point of a deformed region is just its defining site.

Initially, when L is below S, the portion of $V^*(S)$ below L contains no deformed edge. Hence the cross section is empty. As L moves upward the cross section has to be updated, either by starting a new region when L hits a site or by starting a new edge when L hits a vertex that comes from intersecting two deformed separators. The properties of $V^*(S)$ described above will ensure the correctness of this method. Efficiency is gained by organizing points of intersection of L with edges of $V^*(S)$ (the cross section) in a dictionary ordered by x_1-coordinates and by organizing sites and already detected vertices (the critical points) in a priority queue ordered by x_2-coordinates. There are at most $n + (2n - 4)$ critical points since $V(S)$, and thus $V^*(S)$, has at most $2n - 4$ vertices; compare the Introduction. Further, there are at most $n - 1$ points in the cross section since L cannot intersect a region of $V^*(S)$ twice. An update causes only a constant number of operations on both data structures each of which cost $O(\log n)$ time. In conclusion, $V^*(S)$ is computed in $O(n \log n)$ time and $O(n)$ space. It is clear that the original diagram $V(S)$ can be derived from its deformation in $O(n)$ time.

Fortunately, this simple construction method extends to several generalized Voronoi diagrams [Fortune 1987]. It naturally applies to the *additively weighted* type since the correspondence to lower envelopes extends provided cone(p) is translated downward by the weight $w(p)$ for each site p. The intersection of cone(p) with R^2 is then a circle with center p and radius $w(p)$, and indeed this type may be interpreted as the Voronoi diagram for circles under the euclidean distance function (Johnson-Mehl model, Figures 6 and 35). The time bound of $O(n \log n)$ and the space bound of $O(n)$ clearly remains unaffected.

An important type suited for a plane-sweep attack is the Voronoi diagram for noncrossing *straight-line segments* in R^2. The distance of a point x from a segment s is measured by $\min\{\delta(x, y) \mid y \in S\}$. The region of a segment can be deformed so

as to have the lower endpoint of that segment as its bottommost point. Although details are more complicated, a reasonably simple $O(n \log n)$ time and $O(n)$ space algorithm is obtained. In particular, the *medial axis* of a simple polygon is constructable via plane-sweep: The edges of the polygon are taken as line segment sites, and the parts of the diagram outside of the polygon are ignored. Note that the separator of two segments consists of portions of straight lines and parabolas; compare Figure 25.

3.3.3 Constrained Voronoi Diagrams

A further type of Voronoi diagram, and one to which the plane-sweep technique applies particularly nicely, is the classical diagram constrained by a set of line segments [Seidel 1988]. Let S denote a set of n point sites in R^2 and let T be a set of m noncrossing line segments with their endpoints in S. Note that $m = O(n)$ since the segments form a planar graph. We view the segments in T as obstacles and define the *bounded distance* between two points $x, y \in R^2$ as

$$b(x, y) = \begin{cases} \delta(x, y), & \text{if } \overline{xy} \cap T = \emptyset, \\ \infty, & \text{otherwise.} \end{cases}$$

The *b-regions*

$$\mathrm{reg}_b(p) = \{x \in R^2 \mid b(x, p) \le b(x, q),\ q \in S\}$$

for all sites $p \in S$ define the *bounded Voronoi diagram*, $V_b(S, T)$, of S and T. b-regions of endpoint sites are generally nonconvex (near the corresponding obstacle), whereas b-regions of nonendpoint sites have to be convex. Both types of b-regions, and necessarily the former type, might have portions of obstacles rather than of perpendicular bisectors as edges. Clearly, $V_b(S, T)$ is a polygonal cell complex in R^2.

The bounded Voronoi diagram is now modified to the *constrained Voronoi diagram*, $V_c(S, T)$, of S and T by modifying, for each obstacle $t \in T$, the b-regions it supports. Let t support $\mathrm{reg}_b(p_1), \ldots, \mathrm{reg}_b(p_k)$ from the left and $\mathrm{reg}_b(q_1), \ldots, \mathrm{reg}_b(q_l)$ from the right. The former b-regions are extended to the left of t as if only sites $p_1, \ldots, p_k$ were present, and the latter b-regions are extended to the right of t as if only sites $q_1, \ldots, q_l$ were present. Note that $p_i = q_j$ has to occur twice, namely, for the endpoints r and s of t. So $\mathrm{reg}_b(r)$ and $\mathrm{reg}_b(s)$ are extended to both sides of t. The *c-region*, $\mathrm{reg}_c(p)$, of a site $p \in S$ is defined as the union of $\mathrm{reg}_b(p)$ and its extensions with respect to all obstacles. c-regions clearly may overlap, so they fail to define a cell complex and the graph defined by their edges is nonplanar. See Figure 42 illustrating the bounded Voronoi diagram (solid) and the constrained Voronoi diagram (solid or dashed or dotted) for 11 sites and 2 obstacles (bold).

The significance of $V_c(S, T)$ is due to its dual structure, the *constrained Delaunay triangulation* of S and T. Intuitively, this triangulation is as similar as possible to the classical Delaunay triangulation but integrates the obstacles in T, a structure with several practical applications. Formally, it consists of the obstacle segments in T and, in addition, of all edges between sites $p, q \in S$ that have $b(p, q) < \infty$ and that lie on a circle enclosing only sites $r \in S$ with at least one of $b(r, p)$, $b(r, q) = \infty$. The interested reader may verify that two sites are connected in this triangulation if and only if their perpendicular bisector contributes an edge to the constrained Voronoi diagram. Note that the bisector of the endpoint sites of an obstacle may contribute two edges (on either side of the obstacle) rather than only one. Since S contains n sites, T contains m obstacles, and no triangulation of n points can have more than $3n - 6$ edges, we conclude that $V_c(S, T)$ realizes at most $3n + m - 6$ edges and thus has a size of $O(n)$. Figure 27 shows the constrained Delaunay triangulation that corresponds to the constrained Voronoi diagram in Figure 42.

Let us now come to the plane-sweep construction of $V_c(S, T)$. To aid the

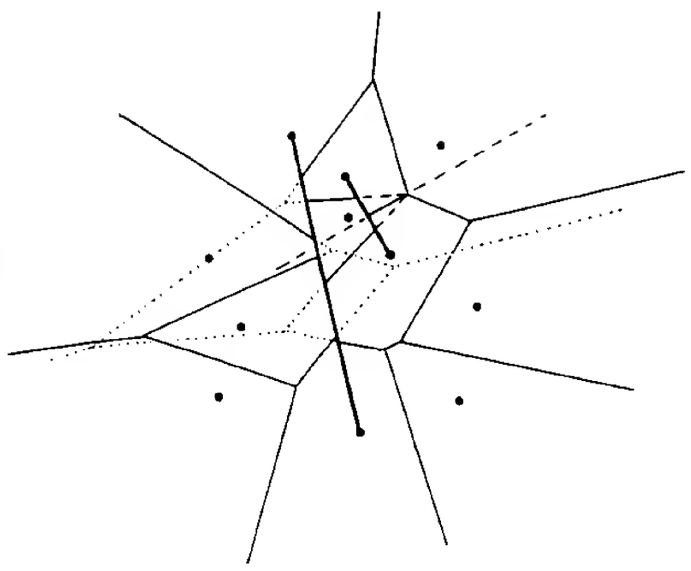


Figure 42. Bounded and constrained Voronoi diagram [Seidel 1988].

intuition, we think of $V_c(S, T)$ as being embedded into $m + 1$ parallel planes in the following way. One plane, call it H_0, contains the bounded Voronoi diagram $V_b(S, T)$. Each of the m planes, H_t, for the obstacle $t \in T$ contains the diagram, V_t, induced by the extensions with respect to t for all b-regions supported by t. Clearly, this embedding comprises the same information as is inherent in $V_c(S, T)$. The idea now is to sweep H_0 and all H_t simultaneously by (essentially one and the same) horizontal line L in order to construct $V_b(S, T)$ and all V_t, for $t \in T$, separately. Again these $m + 1$ diagrams are not constructed directly but via their deformations $V_b^*(S, T)$ and V_t^* that come from mapping upward, within the respective plane, each point x inside an (extension of a) b-region $\mathrm{reg}_b(p)$ by $\delta(x, p)$. Since an extended b-region is just a classical Voronoi region, its deformation has the defining site as the bottommost point (unless the site itself is bottommost for the respective plane). In analogy to the classical Voronoi diagram, regions are created as sites are hit by L, and edges are created as L hits intersections of deformed bisectors. One must, however, take care that edges are assigned correctly to their embedding planes. When a deformed edge in H_0 meets a deformed obstacle t^*, then its construction is continued within H_t and vice versa. The number of critical points arising during the construction of $V_b^*(S, T)$ and all V_t^* is proportional to the size of $V_c(S, T)$ and thus is $O(n)$.

Using appropriate data structures supporting the $m + 1$ plane-sweeps allows us to implement the operations at each critical point in $O(\log n)$ time so an $O(n \log n)$ time and $O(n)$ space algorithm for computing $V_c(S, T)$ is obtained. In particular, the constrained Delaunay triangulation of S and T is computable within these bounds since it can be obtained in $O(n)$ time from $V_c(S, T)$ by exploiting duality.

ACKNOWLEDGMENTS

This survey was completed while the author was at the Institut für Informatik, Freie Universität Berlin, Germany. Work was partially supported by the ESPRIT II Basic Research Program of the EC under contract no. 3075 (project ALCOM).

REFERENCES

AGARWAL, P. K., EDELSBRUNNER, H., SCHWARZKOPF, O., AND WELZL, E. 1990. Euclidean minimum spanning trees and bicromatic closest pairs. In *Proceedings of the 6th Annual ACM Symposium on Computational Geometry*, pp. 203–210.

AGGARWAL, A., AND SURI, S. 1987. Fast algorithms for computing the largest empty rectangle. In *Proceedings of the 3rd Annual ACM Symposium on Computational Geometry*, pp. 278–290.

AGGARWAL, A., HANSEN, M., AND LEIGHTON, T. 1990. Solving query retrieval problems by compacting Voronoi diagrams. In *Proceedings of the 22nd Annual ACM Symposium on STOC*, pp. 331–340.

AGGARWAL, A., GUIBAS, L. J., SAXE, J., AND SHOR, P. W. 1989a. A linear time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete Comput. Geometry 4*, 591–604.

AGGARWAL, A., IMAI, H., KATOH, N., AND SURI, S. 1989b. Finding k points with minimum diameter and related problems. In *Proceedings of the 5th Annual ACM Symposium on Computational Geometry*, pp. 283–291.

AGGARWAL, A., CHAZELLE, B., GUIBAS, L. J., O'DUNLAING, C., AND YAP, C. K. 1988. Parallel computational geometry. *Algorithmica 3*, 293–327.

AHUJA, N. 1982. Dot pattern processing using Voronoi polygons as neighborhoods. *IEEE Trans. Patt. Anal. Mach. Int. PAMI-4*, 336–343.

AKL, S. 1983 A note on Euclidean matchings, triangulations, and spanning trees. *J. Comb. Inf. Syst. Sci. 8*, 169–174.

ALEXANDERSON, G. L., AND WETZEL, J. E. 1978. Simple partition of space. *Math. Magazine 51*, 220–225.

ALT, H., AND YAP, C. K. 1990. Algorithmic aspects of motion planning: A tutorial (part 2). *Algorithms Rev. 1*, 61–77.

AONUMA, H., IMAI, H., IMAI, K., AND TOKUYAMA, T. 1990. Maximin location of convex objects in a polygon and related dynamic Voronoi diagrams. In *Proceedings of 6th Annual ACM Symposium on Computational Geometry*. pp. 225–234.

ARNOLD, D. B., AND MILNE, W. J. 1984. The use of Voronoi tessellations in processing soil survey results. *IEEE Comput. Graph. Appl. 4*, 22–30.

ARONOV, B. 1989. On the geodesic Voronoi diagram of point sites in a simple polygon. *Algorithmica 4*, 109–140.

ARONOV, B., FORTUNE, S., AND WILFONG, G. 1988. The furthest-site geodesic Voronoi diagram. In *Proceedings of the 4th Annual ACM Symposium on Computational Geometry*. pp. 229–240.

ARONOV, B., CHAZELLE, B., EDELSBRUNNER, H., GUIBAS, L. J., SHARIR, M., AND WENGER, R. 1990. Points and triangles in the plane and halving planes in space. In *Proceedings of the 6th Annual ACM Symposium on Computational Geometry*. pp. 112–115.

ASANO, T., AND ASANO, T. 1986. Voronoi diagram for points in a simple polygon. In *Perspectives in Computing 15*, D. S. Johnson, T. Nishizeki, A. Nozaki, H. S. Wilf, Eds. Academic Press, New York, pp. 51–64.

ASANO, T., BHATTACHARYA, B., KEIL, M., AND YAO, F. 1988. Clustering algorithms based on minimum and maximum spanning trees. In *Proceedings of the 4th Annual ACM Symposium on Computational Geometry*, pp. 252–257.

ASH, P. F., AND BOLKER, E. D. 1985. Recognizing Dirichlet tessellations. *Geometriae Dedicata 19*, 175–206.

ASH, P. F., AND BOLKER, E. D. 1986. Generalized Dirichlet tessellations. *Geometriae Dedicata 20*, 209–243.

ASH, P. F., BOLKER, E. D., CRAPO, H., AND WHITELEY, W. 1988. Convex polyhedra, Dirichlet tessellations, and spider webs. In *Shaping Space: A Polyhedral Approach*, M. Senechal and G. Fleck, Eds. Birkhäuser, Boston, pp. 231–250.

AUGENBAUM, J. M., AND PESKIN, C. S. 1985. On the construction of the Voronoi mesh on a sphere. *J. Comput. Phys. 59*, 177–192.

AURENHAMMER, F. 1987a. Power diagrams: Properties, algorithms, and applications. *SIAM J. Comput. 16*, 78–96.

AURENHAMMER, F. 1987b. A criterion for the affine equivalence of cell complexes in R^d and convex polyhedra in R^{d+1}. *Discrete Comput. Geometry 2*, 49–64.

AURENHAMMER, F. 1987c. Recognising polytopical cell complexes and constructing projection polyhedra. *J. Symbolic Comput. 3*, 249–255.

AURENHAMMER, F. 1988a. Improved algorithms for discs and balls using power diagrams. *J. Algorithms 9*, 151–161.

AURENHAMMER, F. 1988b. Linear combinations from power domains. *Geometriae Dedicata 28*, 45–52.

AURENHAMMER, F. 1990a. A new duality result concerning Voronoi diagrams. *Discrete Comput. Geometry 5*, 243–254.

AURENHAMMER, F. 1990b. A relationship between Gale transforms and Voronoi diagrams. *Discrete Appl. Math. 28*, 83–91.

AURENHAMMER, F., AND EDELSBRUNNER, H. 1984. An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern Recognition 17*, 251–257.

AURENHAMMER, F., AND IMAI, H. 1988. Geometric relations among Voronoi diagrams. *Geometriae Dedicata 27*, 65–75.

AURENHAMMER, F., AND SCHWARZKOPF, O. [1991]. A simple on-line randomized incremental algorithm for computing higher-order Voronoi diagrams. In *Proceedings of the 7th Annual ACM Symposium on Computational Geometry*. pp. 142–151.

AURENHAMMER, F., AND STÖCKL, G. 1988. On the peeper's Voronoi diagram. Rep. 264, IIG-TU Graz, Austria.

AURENHAMMER, F., STÖCKL, G., AND WELZL, E. 1991. The post-office problem for fuzzy point sets. Rep. 91-07, FU Berlin, Germany.

AVIS, D., AND BHATTACHARYA, B. K. 1983. Algorithms for computing *d*-dimensional Voronoi diagrams and their duals. *Adv. Comput. Res. 1*, 159–180.

BABENKO, V. F. 1977. On the optimal cubature formulae on certain classes of continuous functions. *Analysis Math. 3*, 3–9.

BADDELEY, A. 1977. A fourth note on recent research in geometrical probability. *Adv. Appl. Probab. 9*, 824–860.

BALTSAN, A., AND SHARIR, M. 1988. On shortest paths between two convex polyhedra. *J. ACM 35*, 267–287.

BENTLEY, J. L., AND MAURER, H. A. 1979. A note on Euclidean near neighboring searching in the plane. *Inf. Process. Lett. 8*, 133–136.

BENTLEY, J. L., WEIDE, B. W., AND YAO, A. C. 1980. Optimal expected-time algorithms for closest-point problems. *ACM Trans. Math. Softw. 6*, 563–580.

BERN, M. W., EPPSTEIN, D., AND GILBERT, J. 1990. Provably good mesh generation. Manuscript, XEROX Palo Alto Research Center, Palo Alto, Calif.

BESAG, J. 1974. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Statist. Soc. B 36*, 192–236.

BIEBERBACH, L. 1912. Über die Bewegungsgruppen der euklidischen Räume I, II. *Math. Ann. 72*, 400–412.

BLUM, H. 1967. A transformation for extracting new descriptors of shape. In *Proceedings of the Symposium on Models for the Perception of Speech and Visual Form*. Weiant Whaten-Dunn, Ed. MIT Press, Cambridge, Mass., pp. 362–380.

BLUM, H. 1973. Biological shape and visual science (Part I). *J. Theor. Biol. 38*, 205–287.

BOISSONNAT, J.-D. 1988. Shape recognition from planar cross sections. *Comput. Vision Graph. and Image Process. 44*, 1–29.

BOISSONNAT, J.-D., AND TEILLAUD, M. 1986. A hierarchical representation of objects: The Delaunay tree. In *Proceedings of 2nd Annual ACM Symposium on Computational Geometry*, pp. 260–268.

BOOTS, B. N. 1979. Weighting Thiessen polygons. *Econ. Geography*, 248–259.

BOOTS, B. N. 1986. Voronoi (Thiessen) polygons. In *CATMOG 45*, Geo Books, Norwich, Conn.

BOWYER, A. 1981. Computing Dirichlet tessellations. *Computer J. 24*, 162–166.

BRASSEL, K. E., AND REIF, D. 1979. A procedure to generate Thiessen polygons. *Geograph. Anal. 11*, 289–303.

BRISSON, E. 1989. Representing geometric structures in d dimensions: Topology and order. In *Proceedings of the 5th Annual ACM Symposium on Computational Geometry*. pp. 218–227.

BRONDSTED, A. 1983. *An Introduction to Convex Polytopes*. Springer, New York, Heidelberg, Berlin.

BROSTOW, W., AND SICOTTE, Y. 1975. Coordination number in liquid argon. *Physica A 80*, 513–522.

BROSTOW, W., DUSSAULT, J. P., AND FOX, B. L. 1978. Construction of Voronoi polyhedra. *J. Comput. Phys. 29*, 81–92.

BROWN, K. Q. 1979. Voronoi diagrams from convex hulls. *Inf. Process. Lett. 9*, 223–228.

BROWN, K. Q. 1980. Geometric transforms for fast geometric algorithms. Ph.D. dissertation, Carnegie-Mellon Univ., Pittsburgh, Penn.

BRUMBERGER, H. AND GOODISMAN. 1983. Voronoi cells: An interesting and potentially useful cell model for interpreting the small angle scattering of catalysts. *J. Appl. Crystallogr. 16*, 83–88.

CALABI, L., AND HARTNETT, W.E. 1968. Shape recognition, prairie fires, convex deficiencies, and skeletons. *Am. Math. Monthly 75*, 335–342.

CANNY, J., AND DONALD, B. 1988. Simplified Voronoi diagrams. *Discrete Comput. Geom. 3*, 219–236.

CHAZELLE, B. 1985. How to search in history. *Inf. Control 64*, 77–99.

CHAZELLE, B., AND EDELSBRUNNER, H. 1987. An improved algorithm for constructing kth-order Voronoi diagrams. *IEEE Trans. Comput. C-36*, 1349–1354.

CHAZELLE, B., AND PREPARATA, F. P. 1986. Half-space range search: an algorithmic application of k-sets. *Discrete Comput. Geom. 1*, 83–94.

CHAZELLE, B., DRYSDALE, R. L., LEE, D. T. 1986a. Computing the largest empty rectangle. *SIAM J. Comput. 15*, 300–315.

CHAZELLE, B., COLE, R., PREPARATA, F. P., AND YAP, C. K. 1986b. New upper bounds for neighbor searching. *Inf. Control 68*, 105–124.

CHAZELLE, B., EDELSBRUNNER, H., GUIBAS, L. J., HERSHBERGER, J. E., SEIDEL, R., AND SHARIR, M. 1990. Slimming down by adding; selecting heavily covered points. In *Proceedings of the 6th Annual ACM Symposium on Computational Geometry*, pp. 116–127.

CHERITON, D., AND TARJAN, R. E. 1976. Finding minimum spanning trees. *SIAM J. Comput. 5*, 724–742.

CHEW, L. P. 1986. There is a planar graph almost as good as the complete graph. In *Proceedings of the 2nd Annual ACM Symposium on Computational Geometry*. pp. 169–177.

CHEW, L. P. 1989a. Constrained Delaunay triangulations. *Algorithmica 4*, 97–108.

CHEW, L. P. 1989b. Guaranteed-quality triangular meshes. Rep. TR 89-983, Dept. Comput. Sci., Cornell Univ., Ithaca, N.Y.

CHEW, L. P., AND DRYSDALE, R. L. 1985. Voronoi diagrams based on convex distance functions. In *Proceedings of the 1st Annual ACM Symposium on Computational Geometry*. pp. 235–244.

CHEW, L. P., AND KEDEM, K. 1989. Placing the largest similar copy of a convex polygon among polygonal obstacles. In *Proceedings of the 5th Annual ACM Symposium on Computational Geometry*. pp. 167–174.

CHOW, A. 1980. Parallel algorithms for geometric problems. Ph.D. dissertation. Dept. Comput. Sci., Univ. of Illinois, Urbana, Ill.

CHRISTOFIDES, N. 1976. Worst-case analysis of a new heuristic for the traveling salesman problem. *Symposium on Algorithms and Complexity*, Carnegie-Mellon Univ., Penn.

CLARKSON, K. L. 1987. New applications of random sampling to computational geometry. *Discrete Comput. Geom. 2*, 195–222.

CLARKSON, K. L. 1989. An algorithm for geometric minimum spanning trees requiring nearly linear expected time. *Algorithmica 4*, 461–468.

CLARKSON, K. L., AND SHOR, P. W. 1988. Algorithms for diametral pairs and convex hulls that are optimal, randomized, and incremental. In *Proceedings of the 4th Annual ACM Symposium on Computational Geometry*. pp. 12–17.

CLARKSON, K. L., AND SHOR, P. W. 1989. Applications of random sampling in computational

geometry, II. *Discrete Comput. Geom. 4*, 387-421.

COLE, R., GOODRICH, M. T., AND O'DUNLAING, C. 1990. Merging free trees in parallel for efficient Voronoi diagram construction. Springer LNCS 443, pp. 432-445.

CONWAY, J. M., AND SLOANE, N. J. A. 1982. Voronoi regions of lattices, second moments of polytopes, and quantization. *IEEE Trans. Inf. Theory IT-28*, 211-226.

CRAIN, L. K. 1978. The Monte Carlo generation of random polygons. *Comput. Geosci. 4*, 131-141.

CRAPO, H. 1979. Structural rigidity. *Struct. Topol. 1*, 13-45.

CRUZ ORIVE, L.-M. 1979. Distortion of certain Voronoi tessellations when one particle moves. *J. Appl. Probab. 16*, 95-103.

DAVID, E. E., AND DAVID, C. W. 1982. Voronoi polyhedra as a tool for studying solvation structure. *J. Chem. Phys. 76*, 4611-4614.

DE FLORIANI, L., FALCIDIENO, B., PIENOVI, C., AND NAGY, G. 1988. On sorting triangles in a Delaunay tessellation. Rep. Inst. Mat. Appl., Consiglio Nazionale della Richerche, Genova, Italy.

DEHNE, F., AND KLEIN, R. 1987. An optimal algorithm for computing the Voronoi diagram on a cone. Rep. SCS-TR-122, School Comput. Sci., Carleton Univ., Ottawa, Canada.

DEHNE, F., AND NOLTEMEIER, H. 1985. A computational geometry approach to clustering problems. In *Proceedings of the 1st Annual ACM Symposium on Computational Geometry*. pp. 245-250.

DELAUNAY, B. N. 1932. Neue Darstellung der geometrischen Kristallographie. *Z. Kristallograph. 84*, 109-149.

DELAUNAY, B. N. 1934. Sur la sphere vide. *Bull. Acad. Science USSR VII: Class. Sci. Math.*, 793-800.

DELAUNAY, B. N. 1963. Theorie der regulären Dirichletschen Zerlegungen des *n*-dimensionalen euklidischen Raumes. *Schriftenreihe Int. Math. Dtsch. Akad. Wiss. Berlin 13*, 27-31.

DELAUNAY, B. N., DOLBILIN, N. P., AND STOGRIN, M. I. 1978. Combinatorial and metric theory of planigons. *Trudy Mat. Inst. Steklov 148*, 109-140.

DEWDNEY, A. K. 1977. Complexity of nearest neighbour searching in three and higher dimensions. Rep. 28, Univ. of Western Ontario, London, Ontario.

DEWDNEY, A. K., AND VRANCH, J. K. 1977. A convex partition of R^3 with applications to Crum's problem and Knuth's post-office problem. *Util. Math. 12*, 193-199.

DILLENCOURT, M. B. 1987a. Traveling salesman cycles are not always subgraphs of Delaunay triangulations. *Inf. Process. Lett. 24*, 339-342.

DILLENCOURT, M. B. 1987b. A non-Hamiltonian, nondegenerate Delaunay triangulation. *Inf. Process. Lett. 25*, 149-151.

DILLENCOURT, M. B. 1987c. Toughness and Delaunay triangulations. In *Proceedings of the 3rd Annual ACM Symposium on Computational Geometry*. pp. 186-194.

DIRICHLET, G. L. 1850. Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *J. Reine u. Angew. Math. 40*, 209-227.

DOBKIN, D. P., AND LASZLO, M. J. 1989. Primitives for the manipulation of three-dimensional subdivisions. *Algorithmica 4*, 3-32.

DOBKIN, D. P., AND LIPTON, R. J. 1976. Multidimensional searching problems. *SIAM J. Comput. 5*, 181-186.

DOBKIN, D. P., FRIEDMAN, S., AND SUPOWIT, K. 1990. Delaunay graphs are almost as good as complete graphs. *Discrete Comput. Geom. 5*, 399-407.

DRYSDALE, R. L. 1990. A practical algorithm for computing the Delaunay triangulation for convex distance functions. In *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms*. pp. 159-168.

DWYER, R. A. 1987. A faster divide-and-conquer algorithm for constructing Delaunay triangulations. *Algorithmica 2*, 137-151.

DWYER, R. A. 1989. Higher-dimensional Voronoi diagrams in linear expected time. In *Proceedings of the 5th Annual ACM Symposium on Computational Geometry*. pp. 326-333.

EDAHIRO, M., KOKUBO, I., AND ASANO, T. 1984. A new point-location algorithm and its practical efficiency: Comparison with existing algorithms. *ACM Trans. Graph. 3*, 86-109.

EDELSBRUNNER, H. 1986. Edge-skeletons in arrangements with applications. *Algorithmica 1*, 93-109.

EDELSBRUNNER, H. 1987. *Algorithms in Combinatorial Geometry*. Springer, Berlin-Heidelberg.

EDELSBRUNNER, H. 1989. An acyclicity theorem for cell complexes in *d* dimensions. In *Proceedings of the 5th Annual ACM Symposium on Computational Geometry*. pp. 145-151.

EDELSBRUNNER, H., AND MAURER, H. A. 1985. Finding extreme points in three dimensions and solving the post-office problem in the plane. *Inf. Process. Lett. 21*, 39-47.

EDELSBRUNNER, H., AND SEIDEL, R. 1986. Voronoi diagrams and arrangements. *Discrete Comput. Geom. 1*, 25-44.

EDELSBRUNNER, H., AND WELZL, E. 1985. On the number of line separations of a finite set in the plane. *J. Combin. Theory Ser. A*, 15-29.

EDELSBRUNNER, H., GUIBAS, L. J., AND SHARIR, M. 1989. The upper envelope of piecewise linear functions: algorithms and applications. *Discrete Comput. Geom. 4*, 311-336.

EDELSBRUNNER, H., GUIBAS, L. J., STOLFI, J. 1986. Optimal point location in a monotone subdivision. *SIAM J. Comput. 15*, 317-340.

EDELSBRUNNER, H., KIRKPATRICK, D. G., AND SEIDEL, R. 1983. On the shape of a set of

points in the plane. *IEEE Trans. Inf. Theory IT-29*, 551-559.

EDELSBRUNNER, H., O'ROURKE, J., AND SEIDEL, R. 1986. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput. 15*, 341-363.

EDELSBRUNNER, H., ROTE, G., AND WELZL, E. 1987. Testing the necklace condition for shortest tours and optimal factors in the plane. *Springer LNCS 267*, 364-375.

EHRLICH, P. E., AND IM HOF, H. C. 1979. Dirichlet regions in manifolds without conjugate points. *Comment. Math. Helvet. 54*, 642-658.

EISELT, H. A., AND PEDERZOLI, G. 1986. Voronoi diagrams and their use: A survey. Part I: Theory; Part II: Applications. In S. Goyal, Ed. *Proceedings of the ASAC 7*, 2. pp. 98-115.

ENGEL, P. 1981. Über Wirkungsbereichsteilungen mit kubischer Symmetrie. *Z. Kristallograph. 154*, 199-215.

EPPSTEIN, D. 1990. Finding the k smallest spanning trees. *Springer LNCS 447*, 38-47.

FAIRFIELD, J. 1979. Contoured shape generation: Forms that people see in dot patterns. In *Proceedings of the IEEE Conference on Systems Man Cybernetics*. pp. 60-64.

FAIRFIELD, J. 1983. Segmenting dot patterns by Voronoi diagram concavity. *IEEE Trans. Patt. Anal. Mach. Int. PAMI-5*, 104-110.

FEDEROFF, E. S. 1885. Elemente der Lehre von den Figuren. *Verh. Russ Min. Ges. St. Petersburg 21*, 1-279.

FIELD, D. A. 1986. Implementing Watson's algorithm in three dimensions. In *Proceedings of the 2nd Annual ACM Symposium on Computational Geometry*. pp. 246-259.

FINNEY, J. L. 1979. Procedure for the construction of Voronoi polyhedra. Note. *J. Comput. Phys. 32*, 137-143.

FORTUNE, S. 1985. A fast algorithm for polygon containment by translation. *Springer LNCS 194*, 189-198.

FORTUNE, S. 1987. A sweepline algorithm for Voronoi diagrams. *Algorithmica 2*, 153-174.

FORTUNE, S. 1988. Sorting helps for Voronoi diagrams. Manuscript, ATT Bell Lab., Murray Hill, N.J.

FRANK, F. C., AND KASPER, J. S. 1958. Complex alloy structures regarded as sphere packings. *Acta Crystallogr. 11*, 184-190.

GABRIEL, K. R., AND SOKAL, R. R. 1969. A new statistical approach to geographic variation analysis. *Syst. Zoology 18*, 259-278.

GAMBINI, R. 1966. A computer program for calculating lines of equilibrium between multiple centers of attraction. Manuscript, Center of Regional Studies, Univ. of Kansas, Lawrence, Kan.

GAREY, M. R., GRAHAM, R. L., AND JOHNSON, D. S. 1976. Some *NP*-complete geometric problems. In *Proceedings of the 8th Annual ACM Symposium on STOC*. pp. 10-22.

GAUSS, C. F. 1840. Recursion der 'Untersuchungen über die Eigenschaften der positiven ternären quadratischen Formen von Ludwig August Seeber. *J. Reine Angew. Math. 20*, 312-320.

GILBERT, E. N. 1962. Random subdivisions of space into crystals. *Ann. Math. Stat. 33*, 958-972.

GOODRICH, M. T., O'DUNLAING, C., AND YAP, C. K. 1989. Constructing the Voronoi diagram of a set of line segments in parallel. *Springer LNCS 382*, 12-23.

GOWDA, I. G., KIRKPATRICK, D. G., LEE, D. T., AND NAAMAD, A. 1983. Dynamic Voronoi diagrams. *IEEE Trans. Inf. Theory IT-29*, 724-731.

GREEN, P. J., AND SIBSON, R. 1977. Computing Dirichlet tesselations in the plane. *Comput. J. 21*, 168-173.

GRUBER, P. M., AND LEKKERKERKER, C. G. 1988. *Geometry of Numbers*. North-Holland Publishing Co., Amsterdam.

GRUBER, P. M., AND RYSKOV, S. S. 1987. Facet-to-facet implies face-to-face. Manuscript.

GRÜNBAUM, B. 1967. *Convex Polytopes*. Interscience, New York.

GRÜNBAUM, B., AND SHEPHARD, G. C. 1980. Tilings with congruent tiles. *Bull. Am. Math. Soc. 3*, 951-973.

GRÜNBAUM, B., AND SHEPHARD, G. C. 1987. *Tilings and Patterns*. Freeman and Co., New York.

GUIBAS, L. J., AND STOLFI, J. 1985. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graph. 4*, 74-123.

GUIBAS, L. J., KNUTH, D. E., AND SHARIR, M. 1990. Randomized incremental construction of Delaunay and Voronoi diagrams. *Springer LNCS 443*, 414-431.

GUIBAS, L. J., MITCHELL, J. S. B., AND ROOS, T. 1991. Voronoi diagrams of moving points in the plane. *Springer LNCS*. In press.

HARTIGAN, J. A. 1975. *Clustering Algorithms*. John Wiley, New York.

HEUSINGER, H., AND NOLTEMEIER, H. 1989. On separable clusterings. *J. Algorithms 10*, 212-227.

HINRICHS, K., NIEVERGELT, J., SCHORN, P. 1988a. Plane-sweep solves the closest pair problem elegantly. *Inf. Process. Lett. 26*, 255-261.

HINRICHS, K., NIEVERGELT, J., SCHORN, P. 1988b. A sweep algorithm for the all-nearest-neighbors problem. *Springer LNCS 333*, 43-54.

HORTON, R. E. 1917. Rational study of rainfall data makes possible better estimates of water yield. *Eng. News-Record*, 211-213.

HOWE, S. E. 1978. Estimating regions and clustering spatial data: Analysis and implementation of methods using the Voronoi diagram. Ph.D. dissertation, Brown Univ., Providence, R.I.

LOEB, A. L. 1970. A systematic survey of cubic crystal structures. *J. Solid State Chem. 1*, 237–267.

MANACHER, G. K., AND ZOBRIST, A. L. 1979. Neither the greedy nor the Delaunay triangulation of a planar point set approximates the optimal triangulation. *Inf. Process. Lett. 9*, 31–34.

MATULA, D. W., AND SOKAL, R. R. 1980. Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geograph. Anal. 12*, 205–221.

MATZKE, E. B., AND NESTLER, J. 1946. Volume-shape relationships in variant foams. *Am. J. Botanics 33*, 130–144.

MAUS, A. 1884. Delaunay triangulation and the convex hull of n points in expected linear time. *BIT 24*, 151–163.

MAXWELL, J. C. 1864. On reciprocal diagrams and diagrams of forces. *Phil. Mag. 4*, 27, 250–261.

MCLAIN, D. H. 1976. Two dimensional interpolation from random data. *Comput. J. 19*, 178–181.

MEGIDDO, N. 1983. Linear time algorithms for linear programming in R^3 and related problems. *SIAM J. Comput. 12*, 759–776.

MEHLHORN, K., O'DUNLAING, C., AND MEISER, S. 1990. On the construction of abstract Voronoi diagrams. *Springer LNCS 415*, 227–239.

MEIJERING, J. L. 1953. Interface area, edge length, and number of vertices in crystal aggregates with random nucleation. *Philips Res. Rept. 8*, 270–290.

MILES, R. E. 1970. On the homogenous planar Poisson process. *Math. Biosci. 6*. 85–127.

MITCHELL, J. S. B., MOUNT, D. M., AND PAPADIMITRIOU, C. H. 1987. The discrete geodesic problem. *SIAM J. Comput. 16*, 647–668.

MOLLISON, D. 1977. Spatial contact models for ecological and epidemic spread. *J. Roy. Stat. Soc. B 39*, 283–326.

MONTANARI, U. 1968. A method for obtaining skeletons using a quasi-Euclidean distance. *J. ACM 15*, 600–624.

MORAN, P. A. P. 1966. A note on recent research in geometric probability. *J. Appl. Probab. 3*, 453–463.

MORAN, P. A. P. 1969. A second note on recent research in geometrical probability. *Adv. Appl. Probab. 1*, 73–89.

MOUNT, D. M. 1985. Voronoi diagrams on the surface of a polyhedron. Rep. 1496, Univ. of Maryland.

MOUNT, D. M. 1986. Storing the subdivision of a polyhedral surface. In *Proceedings of the 2nd Annual ACM Symposium on Computational Geometry*. pp. 150–158.

MOUNT, D., AND SAALFELD, A. 1988. Globally-equiangular triangulation of co-circular points in $O(n \log n)$ time. In *Proceedings of the 4th Annual ACM Symposium on Computational Geometry*. pp. 143–152.

MUDER, D. J. 1988a. How big is an n-sided Voronoi polygon? Manuscript, MITRE Corp., Bedford, Mass.

MUDER, D. J. 1988b. Putting the best face on a Voronoi polyhedron. *Proc. London Math. Soc. 56*, 329–348.

MULMULEY, K. 1991. On levels in arrangements and Voronoi diagrams. *Discrete Comput. Geom.* In press.

MURTAGH, F. 1983. A survey of recent advances in hierarchical clustering algorithms. *Computer J. 26*, 354–359.

NEWMAN, C. M., RINOTT, Y., AND TVERSKY, A. 1983. Nearest neighbor and Voronoi regions in certain point processes. *Adv. Appl. Probab. 15*, 726–751.

NIGGLI, R. 1927. Die topologische Strukturanalyse. *Z. Kristallograph. 65*, 391–415.

NOWACKI, W. 1933. Der Begriff 'Voronoischer Bereich'. *Z. Kristallograph. 85*, 331–332.

NOWACKI, W. 1976. Über allgemeine Eigenschaften von Wirkungsbereichen. *Z. Kristallograph. 143*, 360–368.

O'DUNLAING, C., AND YAP, C. K. 1985. A "retraction" method for planning the motion of a disc. *J. Algorithms 6*, 104–111.

O'DUNLAING, C., SHARIR, M., AND YAP, C. K. 1986. Generalized Voronoi diagrams for moving a ladder: I. Topological analysis. *Comm. Pure Appl. Math. 39*, 423–483.

O'DUNLAING, C., SHARIR, M., AND YAP, C. K. 1987. Generalized Voronoi diagrams for moving a ladder: II. Efficient construction of the diagram. *Algorithmica 2*, 27–59.

OHYA, T., IRI, M., AND MUROTA, K. 1984a. A fast Voronoi diagram algorithm with quaternary tree bucketing. *Inf. Process. Lett. 18*, 227–231.

OHYA, T., IRI, M., AND MUROTA, K. 1984b. Improvements of the incremental methods for the Voronoi diagram with computational comparison of various algorithms. *J. Operations Res. Soc. Japan 27*, 306–337.

OKABE, A., YOSHIKAWA, T., FUJII, A., AND OIKAWA, K. 1988. The statistical analysis of a distribution of active points in relation to surface-like elements. *Environ. Plan. A 20*, 609–620.

OVERMARS, M. 1981. Dynamization of order decomposable set problems. *J. Algorithms 2*, 245–260.

PACH, J., AND SHARIR, M. 1989. The upper envelope of piecewise linear functions and the boundary of a region enclosed by convex plates. *Discrete Comput. Geom. 4*, 291–310.

PAPADIMITRIOU, C. H. 1977. The Euclidean traveling salesman problem is *NP*-complete. *Theor. Comp. Sci. 4*, 237–244.

PASCHINGER, I. 1982. Konvexe Polytope und

HUTTENLOCHER, D. P., KEDEM, K., AND SHARIR, M. 1991. The upper envelope of Voronoi surfaces and its applications. In *Proceedings of the 7th Annual ACM Symposium on Computational Geometry*, 194–203.

HWANG, F. K. 1979. An $O(n \log n)$ algorithm for rectilinear minimal spanning trees. *J. ACM 26*, 177–182.

IMAI, H., IRI, M., AND MUROTO, K. 1985. Voronoi diagram in the Laguerre geometry and its applications. *SIAM J. Comput. 14*, 93–105.

IMAI, K., SUMINO, S., IMAI, H. 1989. Minimax geometric fitting of two corresponding sets of points. In *Proceedings of the 5th Annual ACM Symposium on Computational Geometry*. pp. 266–275.

JOHNSON, W. A., AND MEHL, R. F. 1939. Reaction kinetics in processes of nucleation and growth. *Trans. Am. Instit. Mining Metall. A.I.M.M.E. 135*, 416–458.

KANTABUTRA, V. 1983. Traveling salesman cycles are not always subgraphs of Voronoi duals. *Inf. Process. Lett. 16*, 11–12.

KEIL, J. M., AND GUTWIN, C. A. 1989. The Delaunay triangulation closely approximates the complete Euclidean graph. *Springer LNCS 382*, 47–56.

KIANG, T. 1966. Random fragmentation in two and three dimensions. *Z. Astrophysik 64*, 433–439.

KIRKPATRICK, D. G. 1979. Efficient computation of continuous skeletons. In *Proceedings of the 20th Annual IEEE Symposium on FOCS*. pp. 18–27.

KIRKPATRICK, D. G. 1980. A note on Delaunay and optimal triangulations. *Inf. Process. Lett. 10*, 127–128.

KIRKPATRICK, D. G. 1983. Optimal search in planar subdivisions. *SIAM J. Comput. 12*, 28–35.

KLEE, V. 1980. On the complexity of d-dimensional Voronoi diagrams. *Archiv Math. 34*, 75–80.

KLEIN, R. 1989. *Concrete and Abstract Voronoi Diagrams*. Springer LNCS 400.

KLEIN, R., AND WOOD, D. 1988. Voronoi diagrams based on general metrics in the plane. Springer LNCS 294, pp.281–291.

KNUTH, D. 1973. *The Art of Computer Programming III: Sorting and Searching*. Addison-Wesley, Reading, Mass.

KOCH, E. 1973. Wirkungsbereichspolyeder und Wirkungsbereichsteilungen zu kubischen Gitterkomplexen mit weniger als drei Freiheitsgraden. *Z. Kristallograph. 138*, 196–215.

KOPEC, R. J. 1963. An alternative method for the construction of Thiessen polygons. *Professional Geographer 15*, 24–26.

KRUSKAL, J. B. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Problem. Am. Math. Soc. 7*, 48–50.

LANTUEJOUL, C., AND MAISONNEUVE, F. 1984. Geodesic methods in quantitative image analysis. *Pattern Recogn. 17*, 177–187.

LAVES, F. 1930. Ebeneneinteilung in Wirkungsbereiche. *Z. Kristallograph. 76*, 277–284.

LAWSON, C. L. 1972. Generation of a triangular grid with applications to contour plotting. Tech. Memorandum 299, California Inst. Tech. Jet Propulsion Lab.

LAWSON, C. L. 1977. Software for C^1 surface interpolation. In *Mathematical Software III*, J. Rice Ed., Academic Press, New York.

LEE, D. T. 1980. Two-dimensional Voronoi diagrams in the L_p-metric. *J. ACM 27*, 604–618.

LEE, D. T. 1982a. On k-nearest neighbor Voronoi diagrams in the plane. *IEEE Trans. Comput. C-31*, 478–487.

LEE, D. T. 1982b. Medial axis transform of a planar shape. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI-4*, 363–369.

LEE, D. T., AND DRYSDALE, R. L. 1981. Generalization of Voronoi diagrams in the plane. *SIAM J. Comput. 10*, 73–87.

LEE, D. T., AND LIN, A. K. 1986. Generalized Delaunay triangulation for planar graphs. *Discrete Comput. Geom. 1*, 201–217.

LEE, D. T., AND PREPARATA, F. P. 1984. Computational geometry—A survey. *IEEE Trans. Comput. C-33*, 12, 1072–1101.

LEE, D. T., AND SCHACHTER, B. J. 1980. Two algorithms for constructing a Delaunay triangulation. *Int. J. Comput. Inf. Sci. 9*, 219–242.

LEE, D. T., AND WONG, C. K. 1980. Voronoi diagrams in the $L_1(L_\infty)$ metrics with 2-dimensional storage applications. *SIAM J. Comput. 9*, 200–211.

LEE, D. T., AND YANG, C. C. 1979. Location of multiple points in a planar subdivision. *Inf. Process. Lett. 9*, 190–193.

LEVEN, D., AND SHARIR, M. 1986. Intersection and proximity problems and Voronoi diagrams. *Adv. Robotics 1*, 187–228.

LEVIN, D., AND SHARIR, M. 1987. Planning a purely translational motion of a convex robot in two-dimensional space using Voronoi diagrams. *Discrete Comput. Geom. 2*, 9–31.

LINGAS, A. 1986a. The greedy and Delaunay triangulations are not bad in the average case. *Inf. Process. Lett. 22*, 25–31.

LINGAS, A. 1986b. A fast algorithm for the generalized Delaunay triangulation. Manuscript, Univ. of Linköping, Sweden.

LINGAS, A. 1989. Voronoi diagrams with barriers and the shortest diagonal problem. *Inf. Process. Lett. 32*, 191–198.

LINHART, J. 1981. Die Beleuchtung von Kugeln. *Geom. Dedicata 10*, 145–154.

LITTLE, D. V. 1974. A third note on recent research in geometric probability. *Adv. Appl. Probab. 6*, 103–130.

tessellation of the plane by a Voronoi diagram. *J. Operations Res. Soc. Japan 29*, 69–96.

TANEMURA, M., OGAWA, T., AND OGITA, N. 1983. A new algorithm for three-dimensional Voronoi tessellation. *J. Comput. Phys. 51*, 191–207.

TARSKI, A. 1951. A decision method for elementary algebra and geometry. Univ. of California Press, Berkeley, Calif.

THIESSEN, A. H. 1911. Precipitation average for large area. *Monthly Weather Rev. 39*, 1082–1084.

TOKUYAMA, T. 1988. Deformation of merged Voronoi diagrams with translation. Rep. TR-88-0049, IBM Tokyo Research Lab.

TOUSSAINT, G. T. 1980. The relative neighborhood graph of a finite planar set. *Pattern Recogn. 12*, 261–268.

TOUSSAINT, G. T., AND BHATTACHARYA, B. K. 1981. On geometric algorithms that use the furthest-point Voronoi diagram. Rep. 81-3, McGill Univ., Montreal, Quebec, Canada.

TUOMINEN, O. 1949. Das Einflußgebeit der Stadt Turku ins System der Einflußgebiete SW—Finnlands. *Fennia 71-5*, 1–138.

URQUHART, R. 1980. A note on the computation of subgraphs of the Delaunay triangulation. Rep., Univ. of Glasgow, U.K.

VAIDYA, P. M. 1988a. Minimum spanning trees in k-dimensional space. *SIAM J. Comput. 17*, 572–582.

VAIDYA, P. M. 1988b. Geometry helps in matching. In *Proceedings of the 20th Annual ACM Symposium on STOC*. pp. 422–425.

VORONOI, M. G. 1908. Nouvelles applications des parametres continus a la theorie des formes quadratiques. *J. Reine Angew. Math. 134*, 198–287.

WANG, C. A., AND SCHUBERT, L. 1987. An optimal algorithm for constructing the Delaunay triangulation of a set of line segments. In *Proceedings of the 3rd Annual ACM Symposium on Computational Geometry*. pp. 223–232.

WATSON, D. F. 1981. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *Comput. J. 24*, 167–172.

WEAIRE, D., AND RIVIER, N. 1984. Soap, cells, and statistics—random patterns in two dimensions. *Contemp. Phys. 25*, 59–99.

WHITELEY, W. 1979. Realizability of polyhedra. *Structural Topol. 1*, 46–58.

WIDMEIER, P., WU, Y. F., AND WONG, C. K. 1987. On some distance problems in fixed orientations. *SIAM J. Comput. 16*, 728–746.

WIGNER, E., AND SEITZ, F. 1933. On the constitution of metallic sodium. *Phys. Rev. 43*, 804–810.

WILLIAMS, R. E. 1968. Space-filling polyhedron: Its relation to aggregates of soap bubbles, plant cells, and metal crystalites. *Science 161*, 276–277.

YAO, A. C. 1975. An $O(E \log \log V)$ algorithm for finding minimum spanning trees. *Inf. Process. Lett. 4*, 21–23.

YAO, A. C. 1982. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J. Comput. 11*, 721–736.

YAP, C. K. 1987. An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. *Discrete Comput. Geom. 2*, 365–393.

Received February 1989; final revision accepted January 1991.

Dirichletsche Zellenkomplexe. Ph.D. Dissertation, Inst. f. Math., Univ. Salzburg, Austria.

PHILBRICK, O. 1968. Shape recognition with the medial axis transform. In *Pictorial Pattern Recognition*, G. C. Cheng, R. S. Ledley, D. K. Pollock, and A. Rosenfeld, Eds. Washington, D.C.

PREPARATA, F. P. 1977. Steps into computational geometry. Rep. R-760, Coordinated Science Lab., Univ. of Illinois, Urbana, Ill., pp. 23–24.

PREPARATA, F. P., AND HONG, S. J. 1977. Convex hulls of finite sets of points in two and three dimensions. *Commun. ACM 20*, 87–93.

PREPARATA, F. P., AND SHAMOS, M. I. 1985. *Computational Geometry: An Introduction*. Springer, New York.

PRIM, R. C. 1957. Shortest connection networks and some generalizations. *Bell Syst. Tech. J. 36*, 1389–1401.

RAJAN, V. T. 1991. Optimality of the Delaunay triangulation in R^d. In *Proceedings of the 7th Annual ACM Symposium on Computational Geometry*, pp. 357–372.

RHYNSBURGER, D. 1973. Analytical delineation of Thiessen polygons. *Geograph. Anal. 5*, 133–144.

RIPPA, S. 1990. Minimal roughness property of the Delaunay triangulation. *Comput. Aided Geom. Design 7*, 489–497.

ROGERS, C. A. 1964. *Packing and Covering*. Cambridge University Press, London, New York.

ROHNERT, H. 1988. Moving discs between polygons. *Springer LNCS 317*, 502–515.

ROSENBERGER, H. 1988. Order-*k* Voronoi diagrams of sites with additive weights in the plane. Rep. UIUCDCS-R-88-1431, Dept. Comput. Sci., Univ. Illinois, Urbana, Ill.

ROSENKRANTZ, D. J., STEARNS, R. E., AND LEWIS, P. M. 1977. An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput. 6*, 563–581.

ROWAT, P. F. 1979. Representing spatial experience and solving spatial problems in a simulated robot environment. Ph.D. dissertation, Univ. of British Columbia, Canada.

SAKAMOTO, M., AND TAKAGI, M. 1988. Patterns of weighted Voronoi tessellations. *Sci. Form 3*, 103–111.

SCHÖNFLIES, A. 1891. *Kristallsysteme und Kirstallstruktur*. Teubner, Leipzig.

SCHWARTZ, J., AND YAP, C. K. 1986. *Advances in Robotics*. Lawrence Erlbaum Associates, Hillside, N.J.

SCHWARZKOPF, O. 1989. Parallel computation of discrete Voronoi diagrams. *Springer LNCS 349*, 193–204.

SEIDEL, R. 1981. A convex hull algorithm optimal for point sets in even dimensions. Rep. 81-14, Univ. of British Columbia, Vancouver, Canada.

SEIDEL, R. 1982. The complexity of Voronoi diagrams in higher dimensions. In *Proceedings of the 20th Annual Allerton Conference on CCC*. pp. 94–95.

SEIDEL, R. 1985. A method for proving lower bounds for certain geometric problems. In *Computational Geometry*, G. T. Toussaint Ed. pp. 319–334.

SEIDEL, R. 1986. Constructing higher-dimensional convex hulls at logarithmic cost per face. In *Proceedings of the 18th Annual ACM Symposium on STOC*. pp. 404–413.

SEIDEL, R. 1987. On the number of faces in higher-dimensional Voronoi diagrams. In *Proceedings of the 3rd Annual Symposium on Computational Geometry*. pp. 181–185.

SEIDEL, R. 1988. Constrained Delaunay triangulations and Voronoi diagrams with obstacles. In Rep. 260, IIG-TU Graz, Austria, pp. 178–191.

SHAMOS, M. I. 1975. Geometric complexity. In *Proceedings of the 7th Annual ACM Symposium on STOC*. pp. 224–233.

SHAMOS, M. I. 1978. Computational geometry. Ph.D. dissertation, Yale Univ., New Haven, Conn.

SHAMOS, M. I., AND HOEY, D. 1975. Closest-point problems. In *Proceedings of the 16th Annual IEEE Symposium on FOCS*. pp. 151–162.

SHARIR, M. 1985. Intersection and closest-pair problems for a set of planar discs. *SIAM J. Comput. 14*, 448–468.

SIBSON, R. 1977. Locally equiangular triangulations. *Comput. J. 21*, 243–245.

SIBSON, R. 1979. The Dirichlet tessellation as an aid in data analysis. *Scandinavian J. Stat. 7*, 14–20.

SIBSON, R. 1980. A vector identity for the Dirichlet tessellation. *Math. Proc. Camb. Phil. Soc. 87*, 151–155.

SIFRONY, S., AND SHARIR, M. 1986. A new efficient motion-planning algorithm for a rod in polygonal space. In *Proceedings of the 2nd Annual ACM Symposium on Computational Geometry*. pp. 178–186.

SMITH, C. S. 1954. The shape of things. *Sci. Am.* 58–64.

SNYDER, D. E. 1962. Urban places in Uruguay and the concept of a hierarchy. Festschrift: C. F. Jones. *Northwestern University Studies in Geography 6*, 29–46.

STIFTER, S. 1989. An axiomatic approach to Voronoi diagrams in 3D. Manuscript, RISC, Univ. of Linz, Austria.

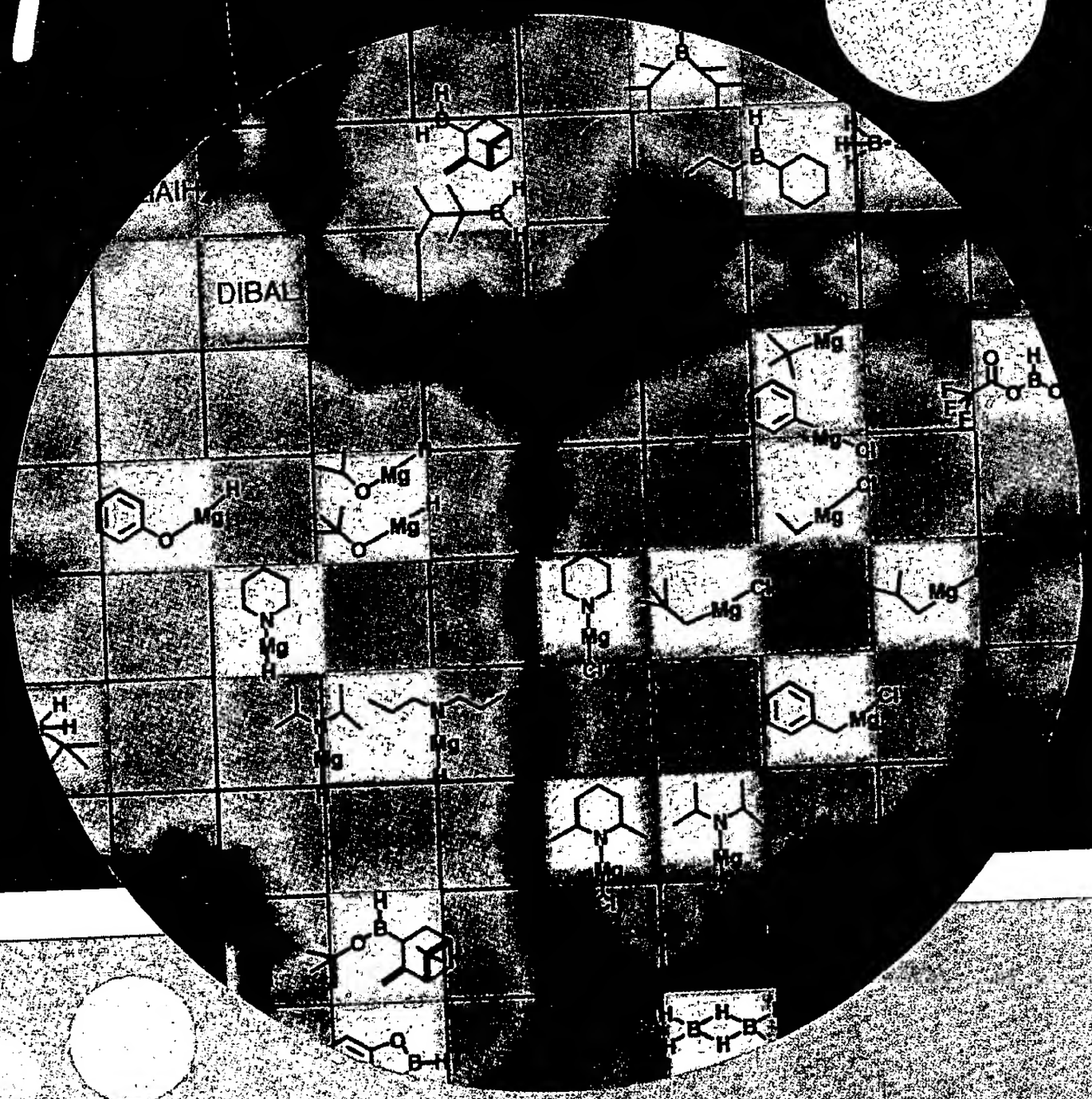
SUGIHARA, K., AND IRI, M. 1988. Geometric algorithms in finite-precision arithmetic. Research Memorandum RMI 88-10, Faculty of Engineering, Univ. of Tokyo, Japan.

SUPOWIT, K. J. 1983. The relative neighborhood graph, with an application to minimum spanning trees. *J. ACM 30*, 428–448.

SUZUKI, A., AND IRI, M. 1986. Approximation of a

EXHIBIT #2

Bulletin of the Chemical Society of Japan



Vol. 73 No. 9
2000

Bull. Chem. Soc. Jpn., 73, 1955—1965 (2000)

Headline Articles

Classification and Prediction of Reagents' Roles by FRAU System with Self-Organizing Neural Network Model

Hiroko Satoh,*,# Kimito Funatsu,† Keiko Takano,†† and Tadashi Nakata

RIKEN (The Institute of Physical and Chemical Research), 2-1 Hirosawa, Wako, Saitama 351-0198

†Department of Knowledge-Based Information Engineering, Toyohashi University of Technology, 1-1 Tempaku, Toyohashi, Aichi 441-8580

††Department of Chemistry, Faculty of Science, Ochanomizu University, 2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610

(Received March 17, 2000)

The classification and prediction of the roles for reagents in reactions are presented. The same dimensional representation of various reagents independent of the number of atoms was achieved by selecting representative factors by the FRAU (Field-characterization for Reaction Analysis and Understanding) system. Training of a self-organizing model considering both negative and absent data was accomplished by modifying the original counter-propagation (CP) type of Kohonen neural network to treat absent data differently from negative data. The modified CP Kohonen neural network successfully classified the reagents and produced a reagent-roles correlation model that gives good answers predicting roles of the reagents.

Reagents play important roles in chemical reactions, and very various reagents have been developed and used in synthetic studies. The roles of reagents may vary according to the substrate, solvent, and so on. In other words, a reagent has various potential roles in reactions. If the roles of the reagents are numerically predicted on a computer before an experiment, it might be very useful in broad fields of synthetic study. For example, the most preferable reagents for the desired reaction can be chosen before synthesis.

In general, reagents in similar structures and electronic features have similar roles, because the potential roles of a reagent are largely related to the features. If sufficient methods to numerate the structural and electronic features and to explain the roles based on the numeric features are possible, numerical prediction of the roles can be achieved.

Recently, we developed the FRAU (Field-characterization for Reaction Analysis and Understanding) system and demonstrated its usefulness to discriminate similarities and differences in structures as well as the roles of metallic reagents based on FRAU's features for metallic atoms and atoms connecting to the metallic atoms.[1,2]

FRAU estimates the possibilities of occurring reactions for a molecule (e.g., a reagent) based on electrostatic and steric interactions with a pseudoreactant (e.g., a substrate). The good points of FRAU are to take account of three-dimensional field around a molecule, and to measure the molecular features, called FRAU features (FFs), based on interactions with a pseudoreactant to analyze chemical reactions. The input data of FRAU are a molecular structure with atomic charges. An arbitrary computational level to obtain the input data is adaptable according to the purposes; namely, the degrees of accuracy and the calculation time are adjustable to the purposes. Thus, FRAU makes it possible to rapidly obtain more accurate properties for molecules than the properties based on topological relationships between atoms.

The number of the FFs, which are estimated for each atom, depends on the number of atoms in a molecule. The atomic features are necessary to know important sites and directions for occurring reactions. However, in order to compare between reagent molecules, representation by the same number of features for various reagents independent of the number of atoms are needed. The same dimensional representation of molecules is one of the important problems in reaction classification studies, where comparing points are needed to discuss the similarities and differences.

In the meanwhile, a neural network is one of appropriate methods for treating multi-dimensional nonlinear data. We used neural networks for reaction classification studies[1,3] because the factors controlling reactions can be considered as data in multi-dimensional space and a reaction resulting from

PRESTO, JST(Japan Science and Technology Corporation).

complicated interactions among these factors can be considered as nonlinear relationships of these multi-dimensional data. Pioneer studies that used neural networks for reactions were performed by Gasteiger et al.[4—9] The first one was an application of a multilayer back-propagation neural network and an associative memory system to learning and predicting the reactivity for a bond.[4,5]

Recently, a self-organizing neural network introduced by Teuvo Kohonen[10—12] has been applied to reaction classification studies, and its usefulness has been demonstrated.[1,3,7—9] The Kohonen neural network makes it possible to classify input data based on the nature being inherent in the input data. This character is suitable for classifying reactions without any preconception of categories or types of reactions as the templates. This is one of the reasons why we have been using the Kohonen neural network for our reaction classification studies.[1,3]

The classification can be developed for predictions by a counter-propagation type of Kohonen neural network (hereafter, called CP Kohonen neural network),[12,13] that is a Kohonen neural network to which known signals and an output layer are added. The processes of characterization, classification and prediction are *formally* similar to the thinking way of a chemist's brain, where, following the recognition and classifications of reactions, predictions are performed. The CP Kohonen neural network gives an organizing map where input data are classified in the basis of the similarities in the input data with a map, where the known signals are trained according to the similarities in the input data. Good points of the CP Kohonen neural network comparing a back-propagation type is to give a stable interpolated answer and to show us major factors of the input data that concern with the interested similarities because the known signals do not influence the classification. The CP Kohonen neural network has an advantage to treat non-linear data that fits our purposes, although there are many statistical methods used to determine major factors, such as principal component analyses. The CP Kohonen neural network trains values for prediction according to similarities in the input data. Thus, if the similarities in the input data have good correspondence with the similarities in the known signals, the CP Kohonen neural network produces a good model that can give an appropriate answer. However, in the original CP Kohonen neural network, which treats one type of known signal, it is not distinguishable between negative and absent data. In order to treat reaction data, it is needed to treat absent data differently from negative ones, e. g., reactions and reagents that did not occur and work, respectively. The negative data are important to consider chemical reactions.

The purpose of this article is to develop and to show a process for the characterization, classification, and prediction of the potential roles of reagents using chemical information with more accuracy than the properties based on the topological relationships between atoms. For this purpose, we first solved problems about the same dimensional representation of reagents based on FFs independent on the number of atoms. Second, we improved the CP Kohonen neural network to treat absent data differently from negative ones. We then have used these results to construct a reagent-roles correlation model that predicts the potential roles of the reagents.

Method

1. Overview. First, a combination of FFs that has a good correspondence with the similarities in the reagents' roles was selected as the representative set of FFs. The selection was performed by searching for a good classification of thirty kinds of reagents (Fig. 1) using the Kohonen neural network.[14] Then, an original counter-propagation (CP) Kohonen neural network was modified to distinguish between absent and negative data. Using the modified CP Kohonen neural network, a self-organizing model that relates similarities in the representative FFs of the thirty kinds of reagents with the roles was constructed. The reproduction and prediction of roles by the model were examined for the reagents and ten kinds of reagents as test data (Fig. 2). Detailed procedures are described below.

2. Model Structures. For the first purpose that is representation of molecules in various structures by the same number of features, boranes, hydrides, Grignard reagents, and bases shown in Fig. 1 were used as the models. The second purpose is to predict the potential roles of the reagents from the reagents' features by a chemical informational approach. The mechanism analyses of reactions for reagents with specific substrates under specific conditions (e.g., solvents, temperature, concentration) are not the purposes here. The structures for some of the reagents in Fig. 1 have been studied by X-ray, NMR, and theoretical analyses.[15—20] Boranes are generally known as dimeric structures, and it is considered that the active species in hydroborations are monomeric structures. Aluminium hydrides, such as diisobutylaluminium hydride (DIBAL) (**15**), are also considered to be monomeric active species in hydroaluminations. Grignard reagents exist as an equilibrium mixture which may involve various mono-, bi-, and poly-nuclear solvated component in solutions, that are known as Schlenk equilibrium, and active species have not been determined because they are varied by the structures of the Grignard reagents, solvents, concentration, and temperature. In this study, simple stable structures in the ground state were analyzed, because if the features of simple structures are well concerned with the roles in reactions, the simple features can be considered to be essential to determine the roles. If the simple features are not sufficient to explain the roles, this means that more complicated features are needed, which will be added in the next stage. The structures were optimized by ab initio RHF/3-21G-(*) molecular-orbital calculations[21] to obtain the geometries and charges for FRAU calculations. Ab initio calclulations were used here to obtain reliable properties for the consisted metallic atoms.

3. Classification of Reagents. Calculation of FFs. FRAU calculated FFs for the thirty kinds of reagents. FRAU estimates three kinds of FFs: extent of reaction field (FF_{field}), electrostatic feature ($FF_{electro}$), and steric feature (FF_{steric}). $FF_{electro}$ and FF_{steric} were calculated in the basis of interactions with unit charge (+1) and with an sp^3 carbon as probes to detect the features, respectively. The positions of each of the probes are dots that are evenly dispersed on the surface of a sphere drawn around each atom in a molecule. The radius of the sphere is the van der Waals radius in this study, and any radius can be chosen. The detailed procedures of FRAU were described in our previous paper.[1]

Selection of Representative FRAU Features. FFs are calculated to each atom in a molecule. Thus, the number of sets of

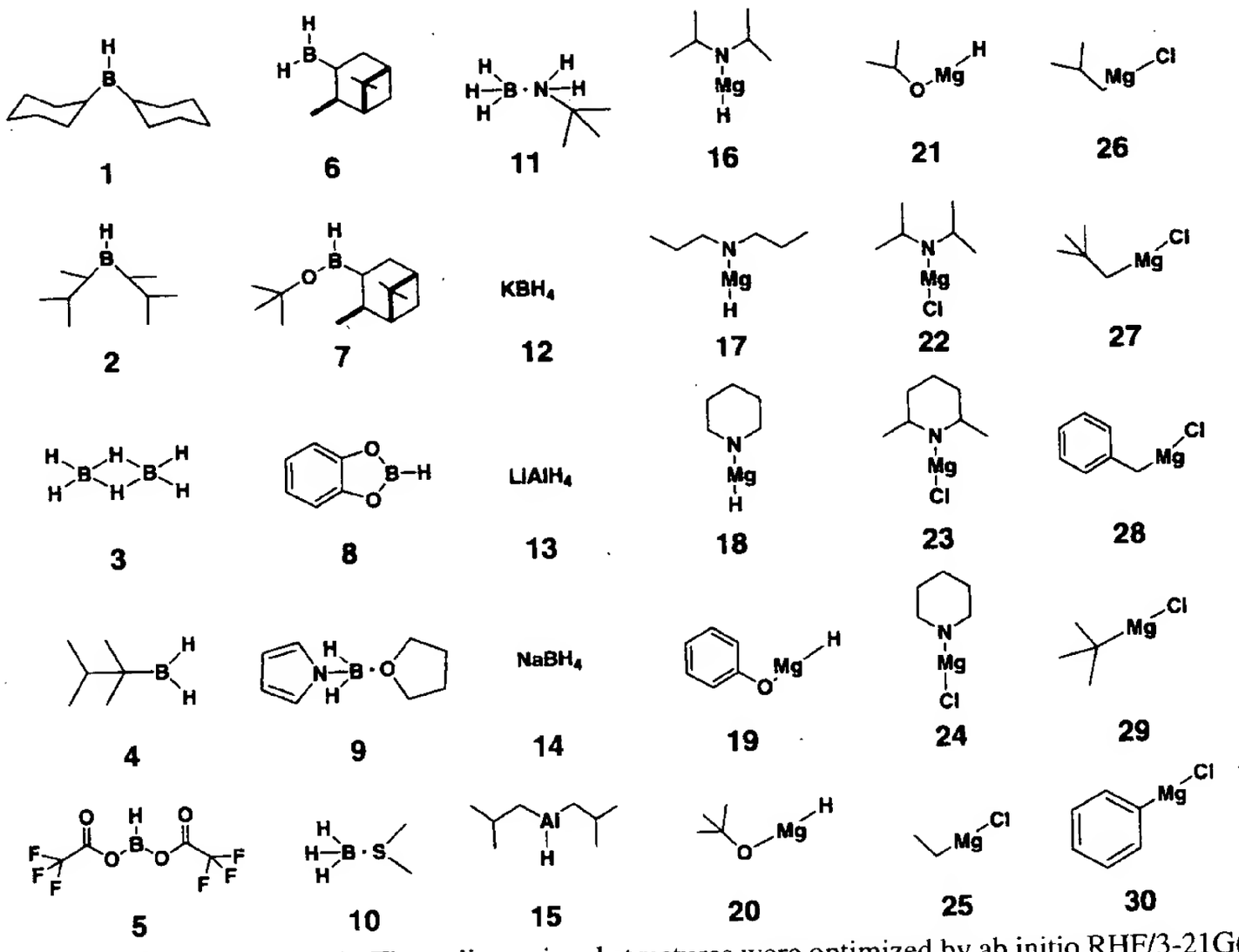


Fig. 1. Model reagents (planar descriptions). Three dimensional structures were optimized by ab initio RHF/3-21G(*) calculations. Optimized geometries for **12**, **13**, and **14** were octahedral type of structures.

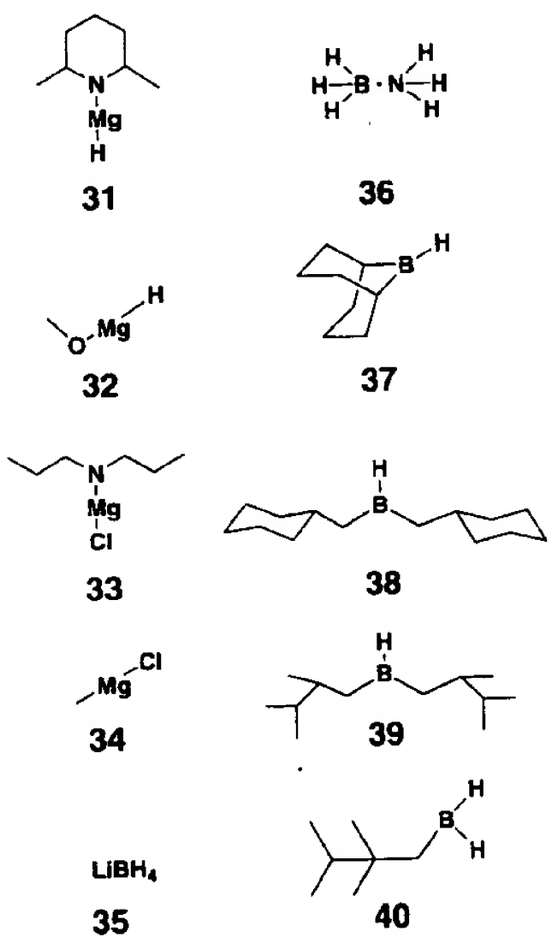


Fig. 2. Reagents for test. They were not used for construction of the model.

the three kinds of FFs is the same as the number of atoms in the molecule. For example, for dicyclohexylborane, which contains 36 atoms ($C_{12}H_{23}B$), 36 sets of the three kinds of FFs are obtained.

In order to compare between reagents in various structures, it is needed to represent the reagents by the same dimensional factors. Thus, the same dimensional representative FFs that have good correspondence with similarities in the structures and roles were selected after investigating various combinations of FFs.

Classification. The selected FFs were input to a Kohonen neural network as discriminators to classify thirty kinds of the reagents (Fig. 1).

The Kohonen neural network outputs a planar map that is called as Kohonen map, where similar input data are set into the same or close neuron, and does not perceive boundaries between clusters of the input data on the Kohonen map. The boundaries were determined by degree of gap between the weight vectors of neighboring neurons by a U-matrix method.[22]

4. Construction of a Reagent-Roles Correlation Model. Counter-Propagation (CP) Kohonen Neural Network. A CP Kohonen neural network is constructed from a Kohonen neural network by the composing of output layers and known answer signals (Fig. 3-(1)). During the training of a CP Kohonen neural network, a modification of the weight vectors between the active and output layers is also performed. In the training, the position of the connection between neurons of active and output layers is determined according to similarities in the input signals, and the dimension and value of the weight vectors between output and active layers are determined according to the signals of a known answer. Weight vectors between output and active layers corresponding to a known answer are trained according to the similarities in the input signals.

Modification of the Original CP Kohonen Neural Network. In order to treat reaction data, negative and absent data should be distinguished; namely, it should be distinguished whether the reaction (reagent) did not occur (work) or the data is just absent from the data set. The distinction is hardly adapted to the original CP Kohonen neural network that has only one type of known answer signals. Thus, the original CP Kohonen neural network was modified according to what follows.[14]

The architecture of the modified CP Kohonen neural network is shown in Fig. 3-(2), where known answer signals for the presence of data are added. The additional signals with the other known answer signals control the training rate of weight vectors between

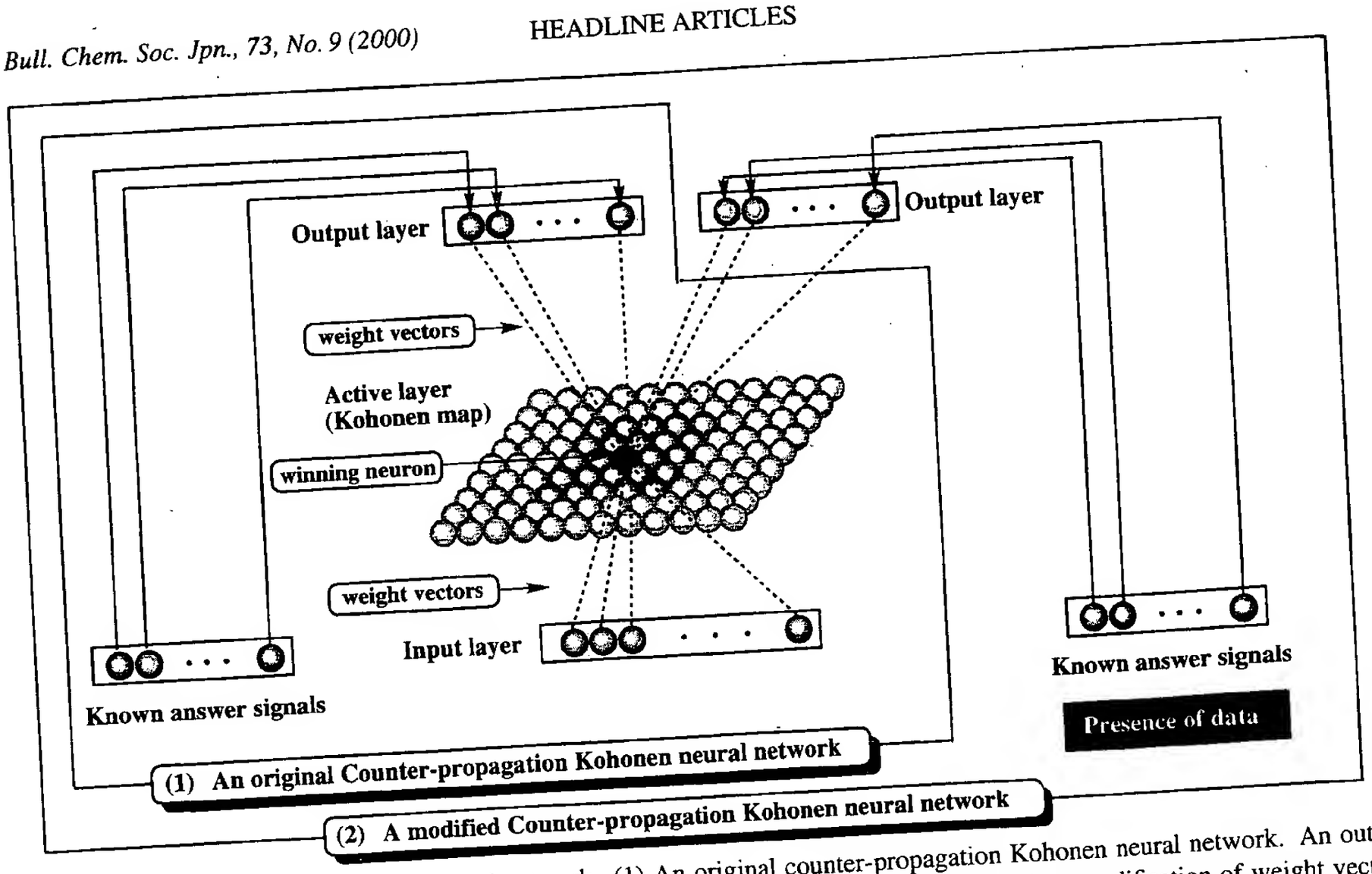


Fig. 3. Counter-propagation Kohonen neural network. (1) An original counter-propagation Kohonen neural network. An output layer and signals of known answer are added to a Kohonen neural network. During the training, modification of weight vectors between the active and output layers is also performed. (2) A counter-propagation Kohonen neural network modified and used, here. A known signal and an output layer for presence of data were added and used in training weight vectors between the active and the output layers.

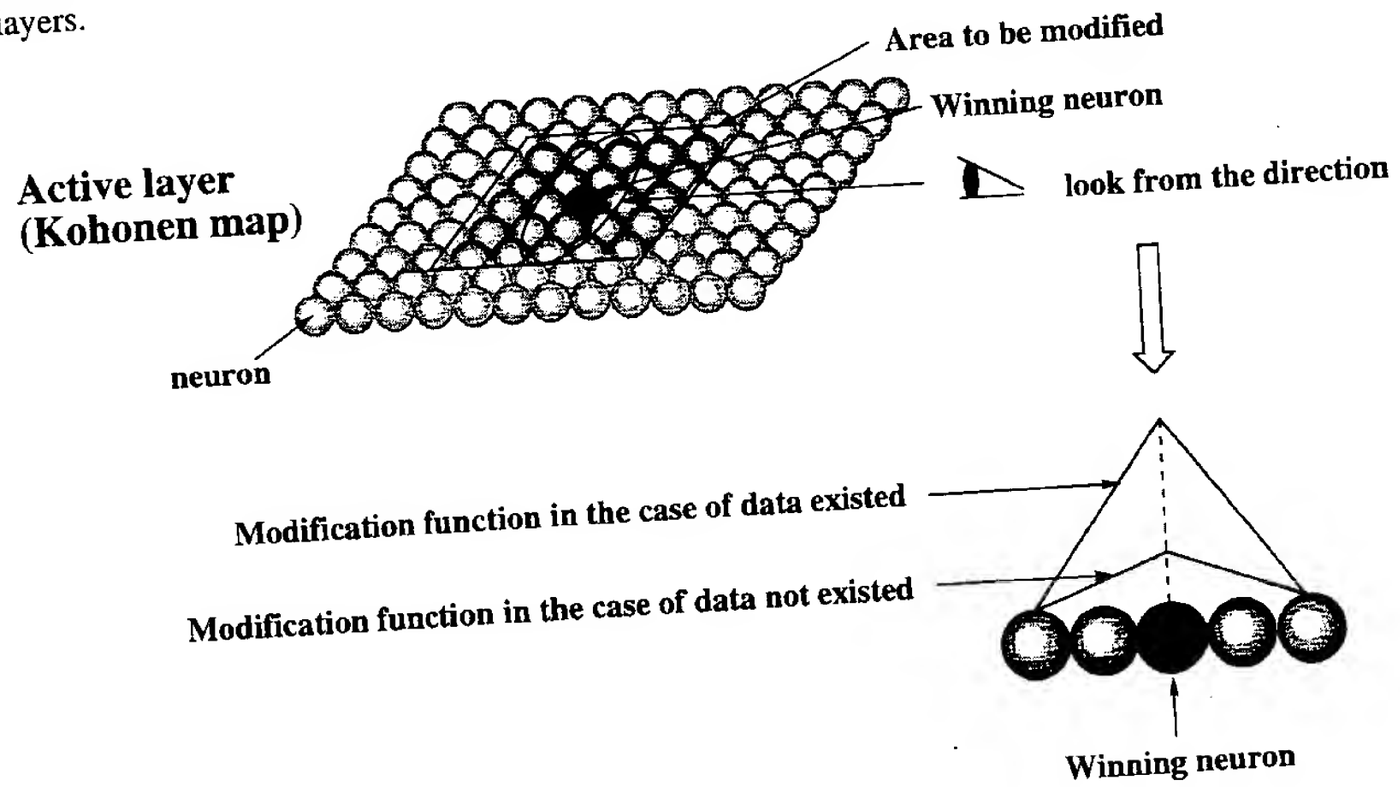


Fig. 4. How to introduce absent data. A rate of the modification function for weight vectors between active and output layers is reduced for the absent data. In this case, triangle modification function was used.

the active and output layers. As shown in Fig. 4, where a triangle function is used as a modification function; when the data is absent, the training rate for weight vectors is reduced to small rather than that for the existed data. Namely, a small voice is given to the absent data to express the possibilities of whether the data might be absent because the reaction (reagent) does not actually occur (work). The volume of the voice for absent data is one of the input parameters.

Construction of the CP Kohonen Neural Network Model. The modified CP Kohonen neural network was used to construct a correlation model between reagents and the roles in reactions. In the construction, the selected representative FFs as the input signals, the roles of the reagents as the first known answer signals, and the information on the presence of the data as the second known answer signals were input. As the known answer signals, nine types of roles of the reagents were set with the information on the presence of the data that were represented by binary notations (1 or 0), as shown in Table 1. All of the binary notations for each reagent are listed in Table 2. The input data for the reagents' roles were determined from some literature and books,[23,24] as well as SYNLIB[25] and ISIS[26] databases. When the working of a reagent against the same functional group varies, a binary notation was determined as follows. If an additional reagent is necessary for

Table 1. Binary Notations Used as Input and Known Signals in the Training of CP Kohonen Neural Network

	There is data where the reagent worked.		There is data where the reagent did not work.		There is no data.	
	Known signal	Presence of data	Known signal	Presence of data	Known signal	Presence of data
Reduction of ketone to alcohol	1	1	0	1	0	0
Reduction of aldehyde to alcohol	1	1	0	1	0	0
Reduction of carboxylic acid to alcohol	1	1	0	1	0	0
Reduction of ester to aldehyde	1	1	0	1	0	0
Reduction of ester to alcohol	1	1	0	1	0	0
Reduction of epoxide	1	1	0	1	0	0
Hydroboration or hydroalumination	1	1	0	1	0	0
Base	1	1	0	1	0	0
Alkylation	1	1	0	1	0	0

Table 2. Input Signals for Thirty Reagents in the Training of Kohonen Neural Network[a,b]

Reagent No.	Reduction of ketone to alcohol		Reduction of aldehyde to alcohol		Reduction of carboxylic acid to alcohol		Reduction of ester to aldehyde		Reduction of ester to alcohol		Reduction of epoxide		Hydroboration or hydroalumination		Base		Alkylation	
	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
2	1	1	1	1	0	1	0	1	0	1	0	0	1	1	0	0	0	0
3	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0
4	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0
5	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
7	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	0	0	0	1	0	1	0	0	1	1	0	0	0	0
9	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
10	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0
11	1	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0
12	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
13	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0
14	1	1	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
16	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
17	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
18	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
19	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
25	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
26	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
28	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
29	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
30	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

a) Reagents' numbers that correspond to those in Fig. 1. b) Roles in reactions. (1) Known answer signals for the roles of reagents. (2) Known answer signals for presence of data.

an interested reagent to work, such as the reduction of epoxides under $NaBH_4$ with $BF_3 \cdot Et_2O$, it was not determined whether the reagent works, because in such a case the interested reagents are changed to the actual active species. If an interested reagent works, even against a specific substrate, it was determined that the reagent works. The volume of the voice for absent data was set at 30% of that for the existing data.

5. Prediction of Roles of Reagents. Reproduction and prediction by the constructed model were investigated for ten reagents used in the modeling (structures of **4**, **6**, **8**, **10**, **13**, **15**, **16**, **21**, **23**, and **27**), and for ten reagents not used in the modeling as test data (Fig. 2), respectively. Reagents **38**, **39**, and **40** have not been

reported, and for the others, experiments have been reported.

Results and Discussion

1. Results of the Classification. A set of six parameters consisting of maximum and minimum $FF_{electro}$ values in a molecule with values of FF_{field} and FF_{steric} on the sites having the maximum and minimum $FF_{electro}$ successfully distinguished similarities and differences in the reagents and was selected as a representative set of FFs. The selected representative sites and the FFs for each molecule are shown in Fig. 5 and Table 3, respectively.

The selected set of FFs was used as discriminators to classify the reagents by the Kohonen neural network. The resulting Kohonen map is shown in Fig. 6-(I). In Fig. 6-(I), each square is a neuron. Reagents were set on neurons where they were drawn. The black solid lines are boundaries recognized by the U-matrix method. The bold solid lines mean a larger difference between neurons than the fine solid lines, and the fine solid lines mean a larger difference between the neurons than the dotted lines. The actual shape of this map is a torus. For visualization, the torus is cut along two perpendicular lines and the surface is spread into a plane. Thus, the top of this map connects with the bottom, and the left edge connects with the right one.

On the Kohonen map, input data having similar FFs are set into the same or close neuron. Thus, the correlation between the FFs and types of the reagents can be found by labeling the map according to the similarities in the types. Figure 6 is colored according to the similarities in the roles of the reagents. It shows that reagents which play similar roles were successfully set on the same or close neurons forming clusters. Borane-dimethyl sulfide (BMS) (**10**) was isolated from cluster **b**, colored grayish blue. The location of BMS is suitable because the roles of BMS are rather similar to those of boranes (cluster **a** colored light blue), even though BMS belongs to borane complexes. The results show that the selected set of FFs has a good correspondence with the similarities of the roles of the reagents.

The reason why several combinations of maximum and minimum values of FFs were mainly investigated for selecting representative FFs is because it is considered that those extreme values might have a higher influence to determine the characters of the reagents. A good correspondence of the six parameters (the maximum and minimum $FF_{electro}$ values in a molecule and values of FF_{field} and FF_{steric} on the sites having the maximum and minimum $FF_{electro}$) with the roles shows that those sites having the maximum and minimum electrostatic interaction energy are major for determining the reagents' characters. Then, in order to know which are the major parameters to discriminate the similarities and differences in the reagents, neurons are colored according to the size of weight vectors corresponding to the six parameters, as shown in Fig. 7. Referring to the maps in Fig. 7 with the classification results in Fig. 6 let us know that the major factors to determine the boundaries are the maximum and minimum of $FF_{electro}$ values. Various combination of FFs were investigated to select the representative FFs, and a set of the maximum and minimum $FF_{electro}$ values was not sufficient to clearly discriminate the similarities, although it was also investigated. This means that the steric interaction

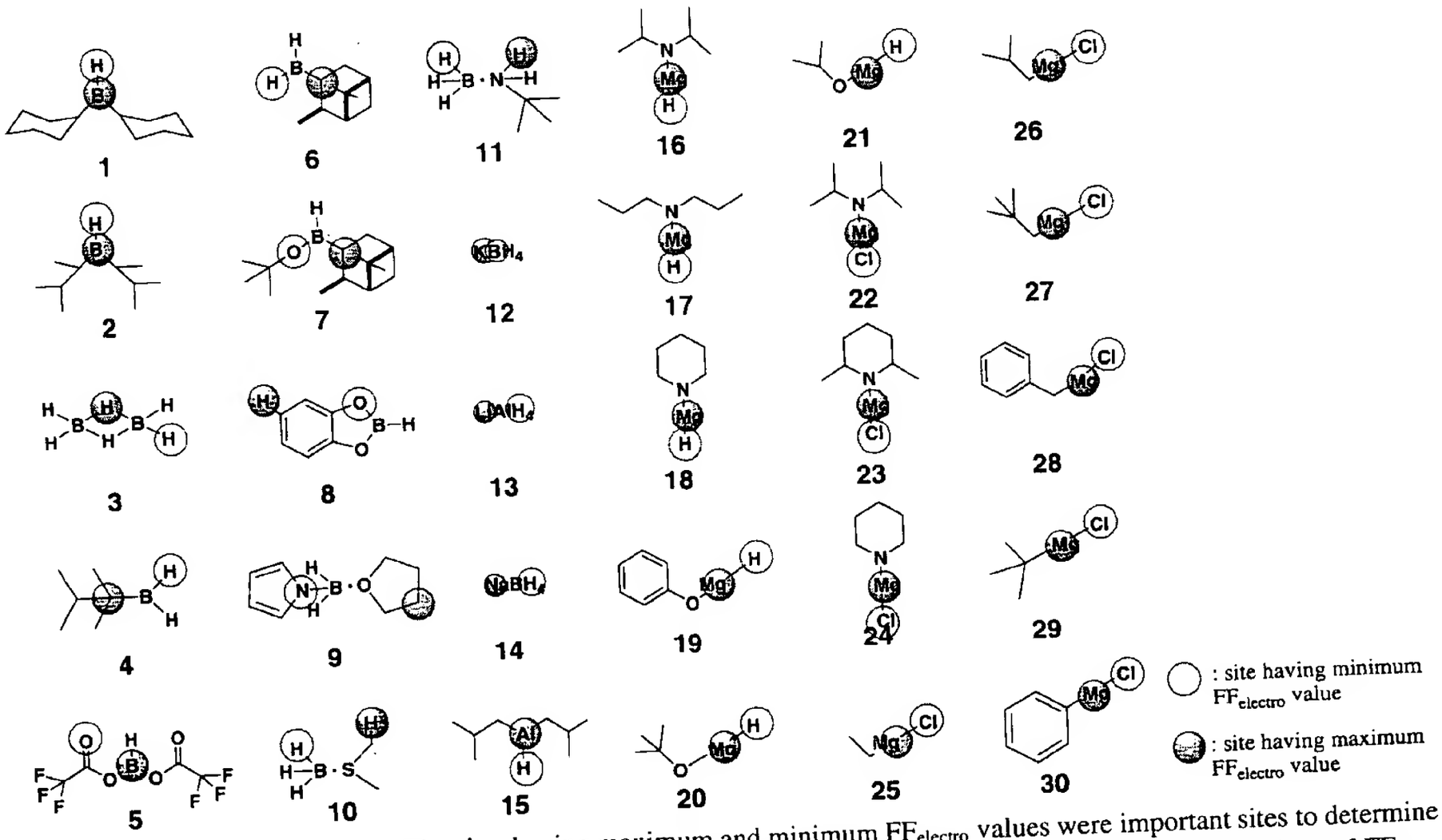


Fig. 5. Selected representative sites. The sites having maximum and minimum $FF_{electro}$ values were important sites to determine the reagents' characters. A set of six parameters of maximum and minimum $FF_{electro}$ values in a molecule with values of FF_{field} and FF_{steric} on the sites having the maximum and minimum $FF_{electro}$ were selected as discriminators that distinguishes similarities and differences in the reagents' roles.

Table 3. Selected Representative FRAU's Features (FFs) for the Reagents

Reagent No.	Sites having minimum $FF_{electro}$			Sites having maximum $FF_{electro}$		
	$FF_{electro}$	FF_{steric}	FF_{field}	$FF_{electro}$	FF_{steric}	FF_{field}
1	−13.454	256.961	253068	10.919	281.860	309377
2	−9.1724	271.503	252849	12.776	292.972	292352
3	−5.2507	267.347	248989	13.274	313.062	151018
4	−11.480	238.851	251560	14.973	477.965	18533
5	−8.1680	190.646	524012	44.590	203.644	438130
6	−11.526	235.594	251559	7.5380	424.580	48907
7	−33.422	222.098	284007	15.621	285.051	216902
8	−26.279	215.623	326616	16.355	261.958	238697
9	−35.407	430.112	68969	24.953	304.273	206290
10	−19.256	260.357	250266	14.445	262.250	233260
11	−39.802	269.070	256267	35.413	258.575	258254
12	−67.907	275.703	370214	26.285	19.6603	2422772
13	−29.474	239.217	257019	41.683	99.0511	847636
14	−55.343	261.024	253623	35.444	39.2887	1526515
15	−20.992	241.131	328616	29.007	265.351	552630
16	−40.573	231.209	362756	26.975	250.520	678728
17	−40.753	230.966	362876	25.117	250.187	704083
18	−36.978	230.305	362680	33.482	228.767	779961
19	−29.897	230.305	360838	42.365	211.166	773137
20	−35.948	230.312	362154	32.833	211.975	772059
21	−36.011	230.151	361815	32.636	211.224	771967
22	−25.630	209.678	946002	38.978	271.667	594481
23	−25.464	209.524	945928	38.632	271.077	615552
24	−22.381	208.736	945287	46.611	244.503	704713
25	−23.125	208.351	946616	50.338	224.836	703261
26	−24.032	208.623	946875	49.068	236.137	680578
27	−24.554	208.811	946937	48.018	246.892	670890
28	−25.820	209.217	946016	42.793	241.687	642569
29	−23.023	208.623	946772	53.404	237.187	692784
30	−20.703	208.659	945317	51.967	231.063	698610

a) Reagents' numbers that correspond to those in Fig. 1.

energy and field extension around the sites are also important to determine characters of the reagents as well as the electrostatic interaction energies on the sites.

2. Results from the Reproduction and Prediction. Reagents Used in the Modeling. The predicted degrees of the roles for ten reagents used in the training of the CP Kohonen neural network model are given in Table 4 along with the input binary notations that stand for the roles and presence of the data. In Table 4, a value that is described on values in parenthesis is the predicted one; the first and second values in parenthesis are the input signal for the roles and for presence of the data, respectively. The range of predicted values was from 0.0 to 1.0. The values of 0.0 and 1.0 correspond to the lowest and highest possibilities to react, respectively.

Almost all of the input values were successfully reproduced, as shown in Table 4, where the predicted values that are largely different from the input ones are underlined. This shows the good capability of the model to reproduce the input values. The underlined cases are discussed below.

The predicted values for roles of isopinocamphan-3-ylborane (**6**) as a reducing agent from aldehydes to alcohols and as that from carboxylic acids to alcohols were extremely different from the input signal. The input signal for the role of **6** was 0 and the predicted value was 0.8. Namely, the organizing map predicted 80% of possibility for **6** to work as reducing agents from aldehydes to alcohols and from carboxylic acids to alcohols, although the input signal denoted that there were no data. The predicted degrees of the roles were according to the similarities in the input FFs between **6** and the other reagents, such as 1,1,2-trimethylpropylborane (thexylborane) (**4**), gathered closely in the Kohonen map. The predicted values for **6** are suitable to knowledge that chemists possess.

For $LiAlH_4$ (**13**), the predicted possibility as a reducing agent from esters to aldehydes was 0.4, although the input signal denoted that it does not work. This was caused by nearby position of **13** to that of **15**. Incidentally, it is known that **13** with Et_2NH reduces esters to aldehydes.

For DIBAL (**15**), 1.0 was predicted to reduce carboxylic acid as the possibility. It is known that **15** does not reduce carboxylic acids, except pyrazolecarboxylic acids. In this execution, for a reagent that works even against to a specific substrate, it was determined the reagent works. Thus, this result is an instance of an overestimate. Some extension is needed in the future, where information on the majorities of the negative data, namely, reaction data on **15** not reactive to many carboxylic acids, are also considered in training the

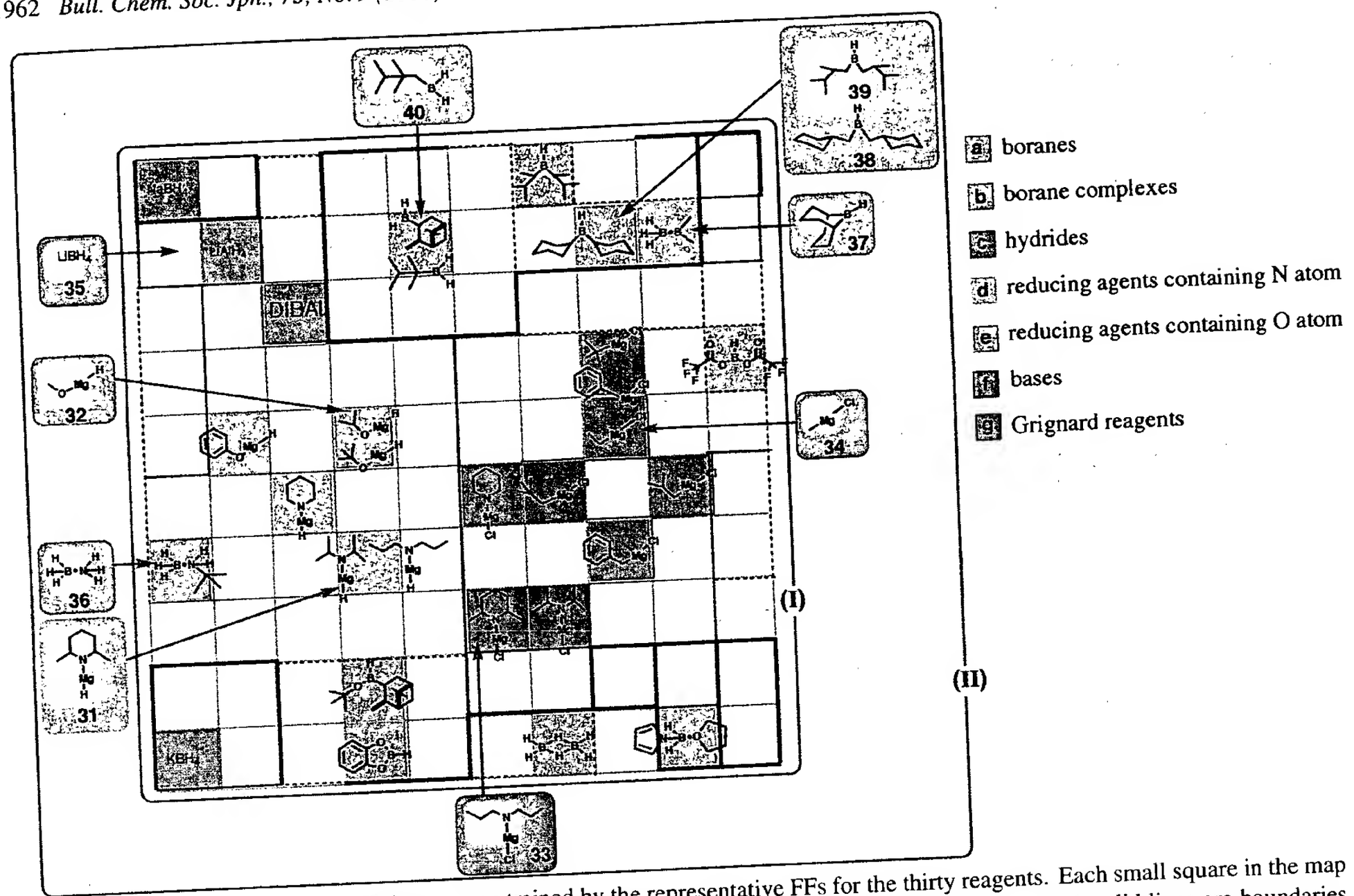


Fig. 6. A Kohonen map. (I) A Kohonen map trained by the representative FFs for the thirty reagents. Each small square in the map is a neuron. Neurons obtaining the input data are colored and mapped by the reagent structures. Black solid lines are boundaries recognized by U-matrix method. Bold solid lines mean larger difference between neurons than the fine solid lines, and the fine solid lines mean larger difference between neurons than the dotted lines. Actual shape of this map is a torus. Thus, the top of this map connects with the bottom, and the left edge connects with the right one. The map is labeled by colors (a—g) according to the roles of the reagents in reactions. (II) The results from mapping of reagents not used in training and constructing of the model. Reagents of **31—40** were mapped on the neurons pointed by arrows.

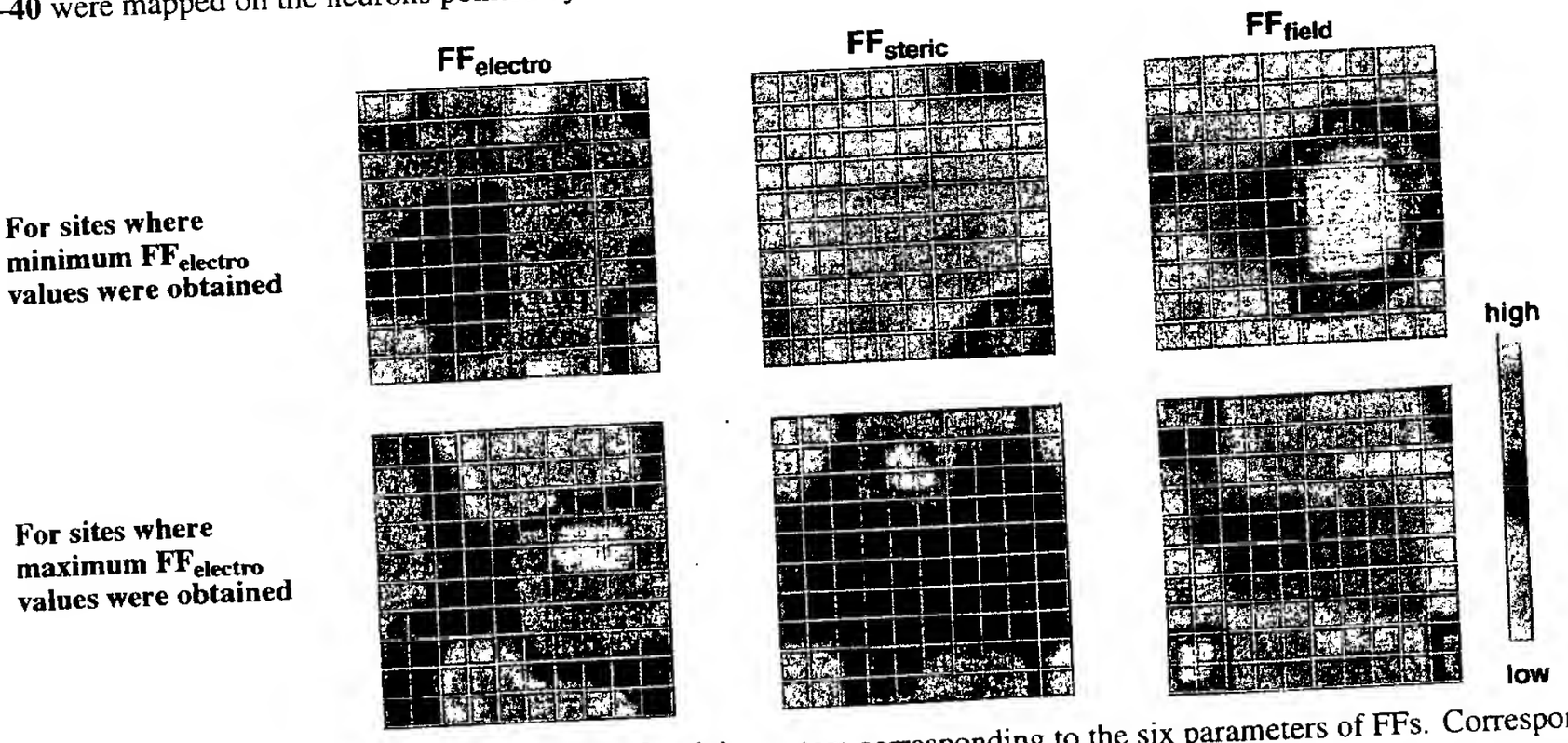
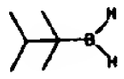
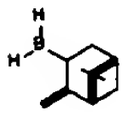
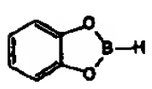
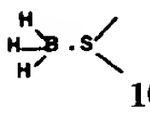
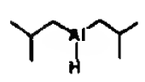
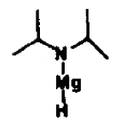
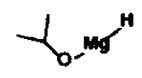
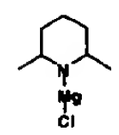
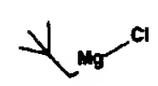


Fig. 7. Kohonen maps colored according to the size of weight vectors corresponding to the six parameters of FFs. Correspondence of gradient between the value and the color on the map are shown in color bar in the right side. Referring of these maps with the classification results in Fig. 6 shows that the major factors to determine the boundaries are maximum and minimum of $FF_{elecuro}$ values.

model. A predicted possibility as a base for **15** was 0.5, but the input signal denoted there was no data. This is suitable to the nature of **15**, which has the basisity.

The predicted value for the roles of neopentylmagnesium chloride (**27**) as base was rather different from the input signal. The input signal for the role for **27** was 0 and the predicted value was 0.6. The predicted degree of the roles was according to the similarities in the input FFs between **27**

Table 4. Results from Prediction of Roles of Reagents Used in Construction of the Model

	Reduction of ketone to alcohol	Reduction of aldehyde to alcohol	Reduction of carboxylic acid to alcohol	Reduction of ester to aldehyde	Reduction of ester to alcohol	Reduction of epoxide	Hydroboration or hydro-alumination	Base	Alkylation
 **4**	1.0 (1,1)	0.8 (1,1)	0.8 (1,1)	0.1 (0,0)	0.1 (0,0)	0.1 (0,0)	1.0 (1,1)	0.0 (0,0)	0.0 (0,0)
 **6**	1.0 (1,1)	0.8 (0,0)	0.8 (0,0)	0.1 (0,0)	0.1 (0,0)	0.1 (0,0)	1.0 (1,1)	0.0 (0,0)	0.0 (0,0)
 **8**	1.0 (1,1)	0.9 (1,1)	0.1 (0,0)	0.0 (0,1)	0.0 (0,1)	0.1 (0,0)	0.9 (1,1)	0.1 (0,0)	0.0 (0,0)
 **10**	1.0 (1,1)	0.9 (1,1)	0.7 (1,1)	0.0 (0,1)	0.7 (1,1)	0.7 (1,1)	1.0 (1,1)	0.0 (0,0)	0.0 (0,0)
$LiAlH_4$ **13**	1.0 (1,1)	1.0 (1,1)	0.8 (1,1)	0.4 (0,1)	1.0 (1,1)	0.8 (1,1)	0.9 (1,1)	0.8 (1,1)	0.0 (0,0)
 **15**	1.0 (1,1)	1.0 (1,1)	1.0 (1,1)	0.8 (1,1)	0.9 (1,1)	0.9 (1,1)	0.9 (1,1)	0.5 (0,0)	0.0 (0,0)
 **16**	1.0 (1,1)	0.0 (0,0)	0.0 (0,0)	0.0 (0,0)	0.0 (0,0)	0.0 (0,0)	0.0 (0,1)	0.0 (0,0)	0.0 (0,0)
 **21**	1.0 (1,1)	0.1 (0,0)	0.1 (0,0)	0.1 (0,0)	0.1 (0,0)	0.1 (0,0)	0.0 (0,0)	0.0 (0,0)	0.0 (0,0)
 **23**	0.1 (0,0)	0.2 (0,0)	0.0 (0,0)	0.0 (0,0)	0.0 (0,0)	0.0 (0,0)	0.0 (0,1)	0.9 (1,1)	0.4 (0,0)
 **27**	0.0 (0,0)	0.0 (0,0)	0.0 (0,0)	0.0 (0,0)	0.0 (0,0)	0.0 (0,0)	0.0 (0,0)	0.6 (0,0)	0.9 (1,1)

and the other base reagents gathered closely in the Kohonen map. This result is also suitable to the nature of **27**, which has the strong basisity, although it is not actually usually used as base.

These results demonstrate that the degrees of similarities in the input FFs have a good correspondence with the degree of similarities in the roles of the reagents, and that the CP Kohonen neural network model was successfully trained.

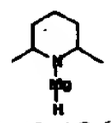
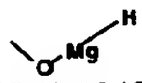
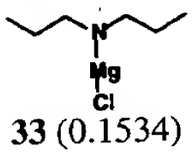
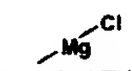
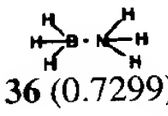
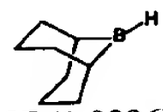
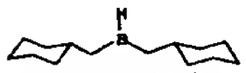
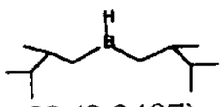
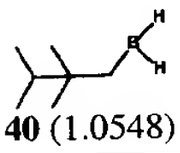
Reagents Not Used in the Modeling. In the CP Kohonen neural network model, ten reagents not used in the modeling (Fig. 2) were allocated as shown in Fig. 6-(II), and the predicted degrees of the roles are listed in Table 5. In the first column of Table 5, the reagent structures and the structural nos. with q-error values in parenthesis are shown. The q-error values indicate a gap between a reagent's FFs and the weight vectors of a neuron where the reagent was mapped. Thus, lower values of the q-error denote a higher reliability of the predicted values.

In Fig. 6, reagents were allocated to the neurons indicated by arrows. The reported borane of 9-borabicyclo-[3.3.1]nonane (9-BBN) (**37**) and ammonia-borane (**36**) are allocated into groups of boranes and borane complexes, respectively. Unreported boranes **38**, **39**, and **40** are allocated into groups of boranes. $LiBH_4$ (**35**) was allocated to a neuron neighbor to $LiAlH_4$. MeOMgH (**32**), 2,6-Me_2-*c*-C_5H_9NMgH (**31**), *n*-Pr_2NMgCl (**33**), and MeMgCl (**34**) were allocated to groups of alkoxymagnesium hydrides, dialkylaminomagnesium hydrides, dialkylaminomagnesium chlorides, and Grignard reagents, respectively. The results show that each of ten reagents was successfully mapped into a suitable neuron.

As shown in Table 5, for **31** and **32**, the possibilities of reducing agents from ketone to alcohol were predicted as 1.0. They were reported as reducing agents from ketone to alcohol.[27] **34**, which is well known as a Grignard reagent, the possibilities of a reagent for alkylation was predicted to be 1.0. Low values of the q-error for them are consistent with these correct results.

For $LiBH_4$ (**35**), all of the predicted values, except that for reducing carboxylic acids, are suitable. The predicted values for reducing carboxylic acids as 0.8, underlined in Table 5, is not good, because it is known that **35** does not reduce carboxylic acids. For 9-BBN (**37**), all of the predicted values, except that for reducing carboxylic acids and esters,

Table 5. Results from Prediction of Roles of Reagents Not Used in Construction of the Model

	Reduction of ketone to alcohol	Reduction of aldehyde to alcohol	Reduction of carboxylic acid to alcohol	Reduction of ester to aldehyde	Reduction of ester to alcohol	Reduction of epoxide	Hydroboration or hydroalumination	Base	Alkyl-ation
 **31** (0.1369)	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
 **32** (0.1242)	1.0	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0
 **33** (0.1534)	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.9	0.4
 **34** (0.1748)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	1.0
$LiBH_4$ **35** (1.0442)	1.0	1.0	0.8	0.0	1.0	0.8	0.9	0.9	0.0
 **36** (0.7299)	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
 **37** (1.0336)	1.0	0.9	0.7	0.0	0.7	0.7	1.0	0.0	0.0
 **38** (0.3696)	1.0	0.8	0.5	0.0	0.4	0.6	1.0	0.0	0.0
 **39** (0.3497)	1.0	0.8	0.5	0.0	0.4	0.6	1.0	0.0	0.0
 **40** (1.0548)	1.0	0.8	0.8	0.1	0.1	0.1	1.0	0.0	0.0

are suitable. The predicted values for reducing carboxylic acids and esters as 0.7, underlined in Table 5, are not good, because it is known that **37** does not reduce them. The q-error values for **35** and **37** were higher. The q-error values denote a lower reliability of the predicted values.

These results show that the predicted values are almost suitable to the potential of the roles, and that the reliability of the predicted values is given as q-error values. Thus, the CP Kohonen neural network model constructed here can predict the roles of those reagents not used in the modeling.

Stability of the CP Kohonen Neural Network Model. Training of the CP Kohonen network model was repeated to the know stability of the model by an examination of the weight vectors between active and output layers. During the training, the weight vectors from about 0.3 to 0.7 were rather unstable, while those from about 0.0 to 0.3 and from about 0.7 to 1.0 were rather stable. The results mean that the predicted values by the model can be used as three or four classes of possibility measures.

A good point of the CP Kohonen neural network is to know the major factors related to the roles of reagents. This is a different good point from a back-propagation (BP) neural network, and the point is the reason to use the CP neural network, here. A weak point of the CP compared with BP neural networks is that the model can not predict the roles for a reagent that is not similar to reagents used in the modeling. Thus, a combination of the good points of CP and BP neural networks will produce a better model for prediction, which is what we are planing to do.

Conclusion

The same dimensional representative features that were calculated by a FRAU system represented similarities in the roles of reagents in various structures. The representative FRAU's features shown that sites where the maximum and minimum electrostatic interaction energies were calculated are important to determine the reagents' characters. The roles of the reagents were successfully classified in the basis of the representative FRAU's features by a Kohonen neural network. The results were applied to a neural network

model construction for predicting the potential roles of the reagents by a CP Kohonen neural network that has been modified to distinguish between negative and absent data. Thus, a reagent-roles correlation model that can predict the possibilities of the reagents' roles was constructed.

The FRAU features represent more detailed similarities and differences in molecules and the FRAU features as discriminators for close structures and roles are now in progress and will be reported in the future. Furthermore, verifications of the predicted values by experiments are also in progress and will be described elsewhere.

The benefits of a CP Kohonen neural network is to treat non-linear data as well as to know factors that principally contribute to the classification. The CP Kohonen neural network produced a reagent-roles correlation model that gives good reproduced and stable interpolated answers. The predicted values can be evaluated in three to four classes. For more accuracy, an improvement of neural network methods will be needed. The reliability of extrapolated answers and the construction of an extended model will be investigated by the CP neural network model with other methods, such as a back-propagation neural network.

The procedures for constructing the reagent-roles correlation model presented here are planned to apply to our reaction prediction studies in the future.

We thank Dr. Kazuo Nagasawa of Synthetic Organic Chemistry Laboratory in RIKEN (The Institute of Physical and Chemical Research) for useful discussions and advice on the chemical reactivity of reagents.

References

1 H. Satoh, S. Itono, K. Funatsu, K. Takano, and T. Nakata, *J. Chem. Inf. Comput. Sci.*, **39**, 671 (1999).

2 Some of the reagents analyzed in the Ref. 1 and here contains boron atoms that do not clearly defined as metallic atoms, however, we call them as metallic atoms, here, to distinguish them from other atoms such as carbon, hydrogen, oxygen, and nitrogen.

3 H. Satoh, O. Sacher, T. Nakata, L. Chen, J. Gasteiger, and K. Funatsu, *J. Chem. Inf. Comput. Sci.*, **38**, 210 (1998).

4 V. Simon, J. Gasteiger, and J. Zupan, *J. Am. Chem. Soc.*, **115**, 9148 (1993).

5 K. Schulz and J. Gasteiger, *J. Chem. Inf. Comput. Sci.*, **33**, 395 (1993).

6 J. Gasteiger and J. Zupan, *Angew. Chem., Int. Ed. Engl.*, **32**, 503 (1993).

7 L. Chen and J. Gasteiger, *Angew. Chem.*, **108**, 844 (1996).

8 L. Chen and J. Gasteiger, *Angew. Chem., Int. Ed. Engl.*, **35**, 763 (1996).

9 L. Chen and J. Gasteiger, *J. Am. Chem. Soc.*, **119**, 4033 (1997).

10 T. Kohonen, *Biol. Cybern.*, **43**, 59 (1982).

11 T. Kohonen, "Self-Organizing Maps," Springer-Verlag, Berlin and Heidelberg (1995).

12 J. Zupan, and J. Gasteiger, "Neural Networks for Chemists. An Introduction," VCH, Weinheim (1993).

13 R. Hecht-Nielsen, *Appl. Optics*, **26**, 4979 (1987).

14 Kohonen neural network and counter-propagation Kohonen neural network programs that were used in this study were modules of TUT_SOM, which is a program package developed by K. Funatsu in Toyohashi University of Technology. TUT_SOM includes two neural network programs (Kohonen neural network and counter-propagation neural network); two programs for recognition of boundaries among weight vectors on Kohonen map (U-matrix and Potential function methods), and several convenient graphic viewers of the results given by the above programs. The modification for treating reaction data was performed for the counter-propagation neural network module of TUT_SOM.

15 a) W. E. Lindsell, in "Comprehensive Organometallic Chemistry," ed by G. Wilkinson, F. G. A. Stone, and E. W. Abel, Pergamon Press, Oxford (1982), Vol. 1, p. 155. b) J. D. Odom, in "Comprehensive Organometallic Chemistry," ed by G. Wilkinson, F. G. A. Stone, and E. W. Abel, Pergamon Press, Oxford (1982), Vol. 1, p. 253. c) W. E. Lindsell, in "Comprehensive Organometallic Chemistry II," ed by G. Wilkinson, F. G. A. Stone, and E. W. Abel, Pergamon Press, Oxford (1995), Vol. 1, p. 57. d) C. E. Housecroft, in "Comprehensive Organometallic Chemistry II," ed by G. Wilkinson, F. G. A. Stone, and E. W. Abel, Pergamon Press, Oxford (1995), Vol. 1, p. 129.

16 a) K. Smith and A. Pelter, in "Comprehensive Organic Synthesis," ed by B. M. Trost, B. I. Fleming, and S. L. Schreiber, Pergamon Press, Oxford (1991), Vol. 8, p. 703. b) J. J. Eisch, in "Comprehensive Organic Synthesis," ed by B. M. Trost, B. I. Fleming, and S. L. Schreiber, Pergamon Press, Oxford (1991), Vol. 8, p. 733.

17 a) P. Chabot, in "Handbook of Grignard Reagents," ed by G. S. Silverman and P. E. Rakita, Marcel Dekker Inc., New York (1996), p. 93. b) J. K. Bonesteel, in "Handbook of Grignard Reagents," ed by G. S. Silverman and P. E. Rakita, Marcel Dekker Inc., New York (1996), p. 103. c) H. L. Uhm, in "Handbook of Grignard Reagents," ed by G. S. Silverman and P. E. Rakita, Marcel Dekker Inc., New York (1996), p. 117.

18 G. H. Penner, Y. C. P. Chang, and J. Hutzal, *Inorg. Chem.*, **38**, 2868 (1999).

19 B. Wrackmeyer, in "Annual Reports on NMR Spectroscopy," ed by G. A. Webb, Academic Press Ltd., London (1988), p. 61.

20 S. Tomoda and T. Senju, *Chem. Commun.*, **1999**, 423.

21 Molecular orbital calculations were performed by SPARTAN system (version 5.0 for Unix) from Wave Function Co., Ltd. Arbitrary computational level of theory is adaptable for geometry optimization and estimation of atomic charges according to purposes of FRAU users, namely, FRAU users can choice a computational level of theory that more rapidly optimizes geometry and estimates atomic charges. We performed ab initio MO calculations, here, to get accurate properties especially for metallic atoms.

22 A. Ultsch, G. Guimaraes, D. Korus, and H. Li, in "Proc. Transputer Anvender Treffen/World Transputer Congress TAT/WTC 93 Aachen," Springer-Verlag, New York (1993), p. 194.

23 L. A. Paquette, in "Encyclopedia of Reagents for Organic Synthesis," John Wiley & Sons, Chichester (1995).

24 A. Pelter, K. Smith, and H. C. Brown, in "Borane Reagents," Academic Press, London (1988).

25 Distributed Chemical Graphics, Inc.

26 Distributed MDL Information Systems, Inc.

27 a) E. C. Ashby and J. J. Lin, *J. Org. Chem.*, **43**, 1557 (1978). b) E. C. Ashby, J. J. Lin, and A. B. Goel, *J. Org. Chem.*, **43**, 1560 (1978). c) E. C. Ashby, J. J. Lin, and A. B. Goel, *J. Org. Chem.*, **43**, 1564 (1978).